



Seija Hänninen-Arffman

## **RUOTSIN HAMMASHUOLLON EFFICAN ANAMNEESIKOM- PONENTIN SUUNNITTELU**

# **RUOTSIN HAMMASHUOLLON EFFICAN ANAMNEESIKOM- PONENTIN SUUNNITTELU**

Seija Hänninen-Arffman  
Opinnäytetyö  
Kevät 2012  
Tietotekniikan koulutusohjelma  
Oulun seudun ammattikorkeakoulu

## TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu  
Tietotekniikan koulutusohjelma, ohjelmistojen kehitys

---

Tekijä(t): Seija Hänninen-Arffman

Opinnäytetyön nimi: Ruotsin hammashuollon Effican anamneesikomponentin suunnittelu

Työn ohjaaja(t): Anne Lahti, Tieto Oyj; Pertti Heikkilä, Oulun seudun ammattikorkeakoulu

Työn valmistumislukukausi ja -vuosi: kevät 2012

Sivumäärä: 47

---

Opinnäytetyön tilaaja on Tieto Oyj:n, Healthcare & Welfare, Effica Healthcare Finland, Nordic Healthcare -yksikkö, jonka toimialueena on Ruotsin julkisen hammashuollon potilastietojärjestelmä Effica.

Tässä opinnäytetyössä suunniteltiin Ruotsin hammashuollon Effica-sovellukseen anamneesikomponentti. Anamneesikomponentti on Effica-sovellukseen liitettävä erillinen komponentti, jossa näkyvät potilaan esitiedot (tiedot sairauksista, lääkityksestä, elämäntavoista ja aikaisemmista hoidoista). Tiedot anamneesikomponenttiin tulevat, kun käyttäjä syöttää tiedot anamneesin käyttöliittymästä. Tulevaisuudessa tiedot voivat tulla myös potilaan täyttämältä web-lomakkeelta. Työhön kuului myös rajapintojen suunnittelu sovelluksen muihin kerroksiin.

Koska asiakkaat toivoivat, että uudessa anamneesissa olisi käytössä Nationell Patientöversiktin Alert Symboli (NPÖ Blomma), perehdyttiin NPÖ-aineistoon ja selvitettiin, miten kyseistä symbolia käytetään. Selvityksen aikana huomattiin, että tiedon välittämisen ja hakemisen suunnittelu NPÖ:stä vaatii enemmän aikaa kuin mitä tähän opinnäytteeseen oli varattu, joten sen toteuttaminen jätettiin myöhemmäksi. Tässä työssä suunniteltiin rakenne, jonka mukaan tiedot NPÖ Blommaan päivitetään anamneesista.

Opinnäytetyön tuloksena on suunnitelma, jonka perusteella anamneesikomponentti voidaan toteuttaa.

---

Asiasanat: anamneesi, terveydenhuollon tietotekniikka, terveydenhuoltoteknologia, Model View ViewModel

## ABSTRACT

Oulu University of Applied Sciences  
Information Technology, Software Development

---

Author(s): Seija Hänninen-Arffman

Title of thesis: Planning of the anamnesis component to Sweden's dental Effica  
Supervisor(s): Anne Lahti, Tieto Oyj; Pertti Heikkilä, Oulun seudun ammattikorkeakoulu

Term and year when the thesis was submitted: Spring 2012

Pages: 47

---

The orderer of this thesis is Tieto Corporation, Health & Welfare, Effica Healthcare, Nordic Healthcare Unit, whose domain business area is the Swedish dental care patient information system Effica.

The objective of this dissertation was to design an anamnesis component to the Effica application of Swedish public dental care. The anamnesis component is a separate component which is connected to the Effica application and in which the patient's medical history (information on diseases, medications, lifestyle and previous treatments) are shown. The information the anamnesis component will come when the user enters the information in the user interface of the anamnesis component. In the future information can be transferred also from the digital anamnesis form filled by the patient. In addition the work included the planning of the application interfaces connected to other layers.

Because the customers wished that there would be Nationell Patientöversikt Alert Symbol (NPÖ Blomma) in use in a new anamnesis, the NPÖ material was studied and was clarified how the symbol is used. During the investigation it was noticed that the planning of the transmitting and searching the information from NPÖ takes more time than time than it had been planned for this thesis so the planning of it was postponed to the later phase. However a structure in which the information for NPÖ Blomma is updated from anamnesis was introduced in this work.

The result of the thesis is a plan for the basis how the anamnesis component can be developed.

---

Keywords: anamnesis, health care information technology, healthcare technology, Model View ViewModel

## **ALKULAUSE**

Tämä opinnäytetyö on tehty Tieto Oyj:n Healthcare & Welfare, Nordic Healthcare -yksikölle. Haluan kiittää kiittää työnohjaajaa R&D Team Manager Anne Lahtea Nordic Healthcare -yksiköstä, ohjaavaa opettajaa tuntiopettaja Pertti Heikkilää ja kielenohjaaja lehtori Tuula Hopeavuorta Oulun seudun ammattikorkeakoulusta sekä Tieto Oyj:n Healthcare & Welfaren työntekijöitä heidän antamastaan avusta ja tuesta opinnäytetyön teossa. Erityiskiitokset ansaitsee myös aviomieheni, joka on ollut kannustava ja hoitanut perheemme arjen pyörittämisen koko opiskelujeni ajan, jotta minä olen voinut keskittyä opintoihini.

Oulussa 11.5.2012

Seija Hänninen-Arffman

# SISÄLLYS

TIIVISTELMÄ	1
ABSTRACT	2
ALKULAUSE	3
SISÄLLYS	4
LYHENTEET JA TERMINOLOGIA	6
1 JOHDANTO	9
2 OHJELMISTOTUOTANTO	10
2.1 Ohjelmiston kehittäminen	10
2.2 Esitutkimus	12
2.3 Määrittely	12
2.4 Suunnittelu	12
3 ANAMNEESIKOMPONENTTI SUUNNITTELUN LÄHTÖKOHDAT	13
3.1 Efficapotentilastietojärjestelmä	13
3.2 Terveydenhuollon tietotekniikka	15
3.3 Terveydenhuollon sovellusten direktiivit ja standardit	16
3.4 Suun terveydenhuollon Effican tuotekuvaus	17
3.4.1 Yleiset lomakkeet Efficassa	18
3.4.2 Yleiset komponentit Efficassa	18
3.4.3 Sähköinen asiointi	19
3.4.4 Tekniset vaatimukset	20
4 ANAMNEESIKOMPONENTIN VAATIMUKSET	21
4.1 Ei-toiminnalliset vaatimukset	21
4.2 Toiminnalliset vaatimukset	22
4.3 Nationell Patientöversikt	23
4.4 Alert Symbol eli NPÖ Blomma	24
4.5 Käytettävyys	27
5 ANAMNEESIKOMPONENTIN TOIMINNALLISET RATKAISUT	29
5.1 Anamneesin avaaminen	30
5.2 Uuden anamneesin luominen	31

5.3 Anamneesin tietojen muokkaaminen	31
5.4 NPÖ Blomma	31
5.5 Tietorivien kysymysten ja otsikoiden lisääminen	32
5.6 Anamneesi sähköiseltä esitietolomakkeelta	32
5.7 Historianäkymä	33
6 ANAMNEESIKOMPONENTIN ARKKITEHTUURISUUNNITELMA	34
6.1 Komponenttipohjainen ohjelmistokehitys	34
6.1.1 Komponentit ja rajapinnat	35
6.1.2 Ohjelmistoarkkitehtuuri	36
6.1.3 Model View ViewModel -arkkitehtuurimalli	37
6.2 Käytettävät teknologiat	39
6.2.1 Open Framework	39
6.2.2 .NET Framework	39
6.2.3 C#	40
6.2.4 WPF	40
6.2.5 XAML	42
6.3 Effica-alusta-arkkitehtuuri	42
6.4 Effica-komponenttiarkkitehtuuri	42
6.5 Anamneesikomponentin arkkitehtuuri	42
7 ANAMNEESIKOMPONENTIN TIETOKANTASUUNNITTELMA	43
7.1 Tietokannat	43
7.2 Relaatiotietokanta ja normalisointi	43
7.3 Microsoft SQL Server ja SQL	44
7.4 Anamneesin tietokanta	44
8 LOPUKSI	46
LÄHTEET	47

## LYHENTEET JA TERMINOLOGIA

<b>.NET Framework</b>	Microsoftin kehittämä ohjelmistokomponenttikirjasto
<b>Anamneesi</b>	Esitiedot, jotka potilas antaa aikaisemmista sairauksistaan sekä nykyisen sairauden alkamisesta ja kulusta
<b>C#</b>	Microsoftin kehittämä ohjelmointikieli, jossa yhdistyy C++:n tehokkuus ja Java-kielen helppokäyttöisyys
<b>CE-merkintä</b>	Valmistaja vakuuttaa tuotteen täyttävän sitä koskevat EU-direktiivien vaatimukset ja tuote on käynyt mahdolliset vaaditut tarkastukset
<b>Code-behind</b>	Tekniikka, jota käytetään web-sivujen ja käyttöliittymien teossa, jossa näkymän ja siihen liittyvän toiminnallisuuden lähdekoodit tallennetaan erillisinä tiedostoina, jolloin näkymän suunnittelijat ja ohjelmoijat voivat työskennellä itsenäisesti
<b>Databinding</b>	Tekniikka, jolla sidotaan kaksi tietolähdettä yhteen ja ylläpidetään tietoja
<b>Effica</b>	Potilastietojärjestelmä, joka on markkinajohtaja Suomessa
<b>Event</b>	Ohjelmiston viestin ilmoittaminen, kun jotain on tapahtunut, esim. näppäimen painamisesta
<b>Fat client</b>	Kaksitasoisen asiakas-palvelin-arkkitehtuurin työasemasovellus
<b>IEEE</b>	Institute of Electrical and Electronics Engineers on kansainvälinen tekniikan alan järjestö, jonka tehtävinä on julkaisutoiminta, tieteellisten konferenssien järjes-



täminen, koulutuksen edistäminen ja alan keskeisten standardien määrittely

### **Kefalometrinen analyysi**

Röntgenlaitteisiin kuuluvalla kefalometrialla tehty mm. pään skannaus, jonka avulla voidaan tutkia pään fysiologiaa

### **Konnektori**

Komponenttien välinen kommunikaatioväylä eli liitin

### **MDD**

Medical Devices Directive 93/42/EEC  
lääkintälaitedirektiivi

### **MVVM**

Model View ViewModel -ohjelmistojen arkkitehtuurimalli

### **NPÖ**

Nationell Patientöversikt eli kansalliset potilasyhteenvetotiedot Ruotsin terveydenhuollossa

### **NPÖ Blomma**

Nationell Patientöversiktin (NPÖ) graafinen varoitus-symboli

### **Olio**

Ohjelmiston perusyksikkö olio-ohjelmoinnissa, joka sisältää joukon loogisesti yhteenkuuluvia tietoja ja toiminnallisuuksia

### **Open framework**

Avoimen lähdekoodin välineistö, jonka avulla voidaan tehdä helposti vaihdettavia ohjelmiston osia ja sovelluksia

### **Parontologia**

Hammaslääketieteen ala, joka tutkii hampaan kiinnityskudoksia: ikeniä, juurisementtiä yms.

### **Plug and Play**

Windows-käyttöjärjestelmissä käytetty termi, jossa laite tai sovellus liitetään tietokoneeseen ja käyttöjärjestelmä itse asentaa tarvittavat ajurit

<b>Pääkäyttäjä</b>	Ylläpitäjä, joka hoitaa sovelluksen perusasetuksia
<b>SQL</b>	IBM:n kehittämä standardoitu kyselykieli, jolla tehdään erilaisia hakuja, muutoksia ja lisäyksiä relaatiotietokantoihin
<b>Sähköinen allekirjoitus</b>	Asiakastietojen eheys, muuttumattomuus ja kiistämättömyys varmistetaan sähköisellä allekirjoituksella tietojen sähköisessä käsittelyssä, tiedonsiirrossa ja säilytyksessä
<b>Tyhmä päätte</b>	eli Thin client saa tarvitsemansa tiedon verkon välityksellä
<b>Vahva tunnistautuminen</b>	Vahva tunnistautuminen perustuu julkisen avaimen menetelmään tai muuhun vahvaan salaukseen. Käytetään mm. verkkotunnistamisen ja -maksamisen mahdollistavissa järjestelmissä
<b>WCF</b>	Windows Communication Foundation muodostaa viestijärjestelmän. WCF yhdenmukaistaa ja luo ohjelmille yhtenäisen käytännön välittää tietoa keskenään
<b>WPF</b>	Windows Presentation Foundation on Windows-versioiden graafinen rajapinta
<b>XAML</b>	Extensible Application Markup Language on kuvauskieli, jolla WPF- ja Silverlight-ohjelmat toteutetaan
<b>Yksikkötesti</b>	Yksittäisten moduulien testaaminen

# 1 JOHDANTO

Tämän opinnäytetyön tavoitteena oli suunnitella Ruotsin hammashuollon Effic-sovellukseen anamneesikomponentti. Anamneesikomponentti on Effic-sovellukseen liitettävä erillinen komponentti, jossa näkyvät potilaan esitiedot (tiedot sairauksista, lääkityksestä, elämäntavoista ja aikaisemmista hoidoista). Tiedot anamneesikomponenttiin kertyvät, kun käyttäjä syöttää tiedot anamneesin käyttöliittymästä, ja tulevaisuudessa tiedot voivat tulla myös potilaan täyttämältä sähköiseltä web-lomakkeelta.

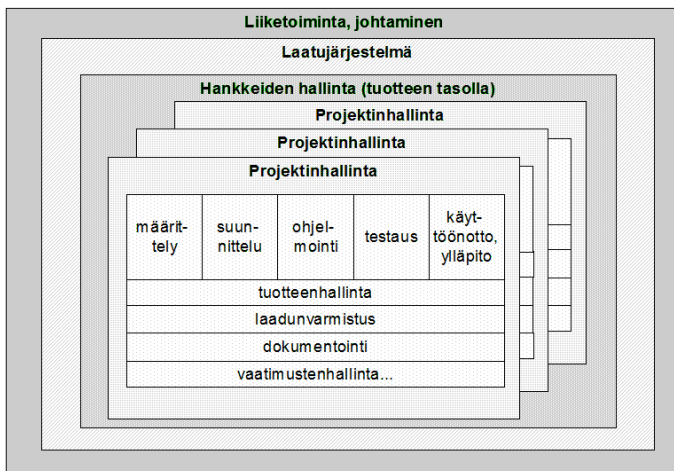
Työhön kuului toiminnallisten ratkaisujen ja käyttöliittymän suunnittelu asiakasvaatimusten mukaisesti sekä tietokanta- ja arkkitehtuurisuunnittelu, johon kuului myös rajapintojen suunnittelu sovelluksen muihin kerroksiin. Yksi työn tärkeimmistä työtehtävistä oli selvittää NPÖ Blomman toimintaperiaate ja suunnitella NPÖ Blomman käyttö anamneesissa tämän toimintaperiaatteen mukaiseksi.

Osa työtuloksista on poistettu tästä opinnäytetyön julkisesta versiosta, koska ne luokitellaan salaiseksi tiedoksi.

## 2 OHJELMISTOTUOTANTO

Ohjelmistotuotanto on suomennettu englannin kielen sanoista ”Software Engineering”. Ohjelmistotuotanto on yhteinen termi työnteossa ja työnjohdossa käytetyille menetelmille tuotettaessa tietokoneohjelmia sekä tietokoneohjelmistoja. Vapaasti tulkittuna termi tarkoittaa ohjelmistotyötä, jonka tuloksena syntyvä järjestelmä toteuttaa käyttäjän kohtuulliset toiveet ja valmistuu tietyn kustannusarvion ja aikataulun mukaisesti. (Haikala – Märijärvi 2006, 16; Ohjelmistotuotanto. 2012.)

Ohjelmistotuotantoon kuuluvat kaikki ohjelmistojen valmistuksen osa-alueet (kuva 1), kuten projektinhallinta, laatujärjestelmät, määrittely, suunnittelu, toteutus, käyttöönotto, laadunvarmistus, testaus, dokumentointi, tuotteen hallinta ja ylläpito. Ohjelmistosuunnittelussa toteutetaan kaikki muut osa-alueet, muttei toteutusta, testausta, käyttöönottoa eikä ylläpitoa. (Haikala – Märijärvi 2006, 16.)



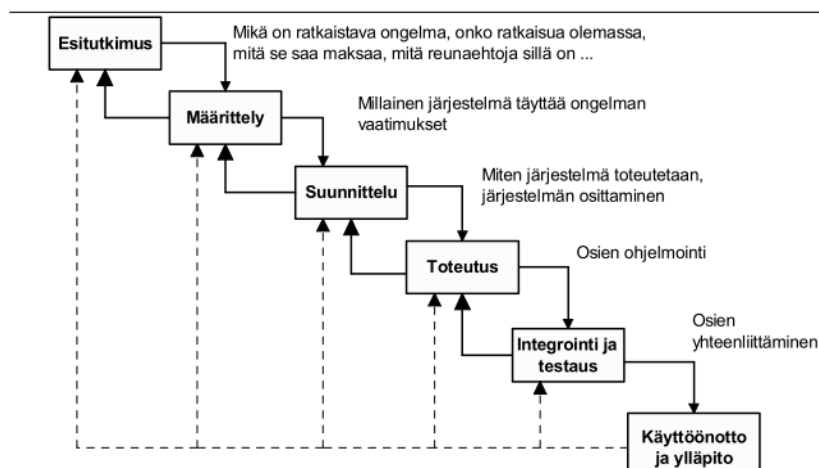
KUVA 1. Ohjelmistotuotannon osa-alueet (Haikala 2006, 35)

### 2.1 Ohjelmiston kehittäminen

Ohjelmistotuotannossa ohjelmistojen kehittäminen toteutetaan yleisemmin projekteissa. Käytännön syistä projekti usein jaetaan peräkkäin tai rinnakkain to-

teuttaviin osaprojekteihin. Yleisesti projektit jaetaan määrittely- ja toteutusprojekteihin. (Haikala – Märijärvi 2006, 53.)

Ohjelmistokehitysprosessien tuottamiselle on olemassa monenlaisia malleja. Mikään ratkaisu ei ole se ainut oikea, vaan se, mikä toimii yhdessä projektissa, ei välttämättä toimi toisessa. Eri vaihejakomallien eli prosessimallien avulla ohjelmistojen kehitystyö tai koko elinkaari jaetaan vaiheisiin. Jokaisella osavaiheella on oma määritelty tuotoksensa (dokumentit, koodi, valmis ohjelma jne.). Jokaiseen eri vaiheeseen liittyy laaduntarkastustoimenpiteitä, kuten tarkastuksia, katselmuksia ja testausta, joilla pyritään pääsemään eroon virheistä mahdollisimman aikaisessa vaiheessa. Käytännössä ohjelmankehitysprosessi on aina iteratiivinen: tulosta tarkennetaan, siirrytään seuraavaan, tullaan uudessa vaiheessa takaisin jne. Vanhin käytetty vaihejakomalli on vesiputousmalli (kuva 2). (Haikala – Märijärvi 2006, 37; Nousiainen 2009; Taina 2000.)



KUVA 2. Vesiputousmalli (Nousiainen 2009)

Tässä projektissa toteutetaan ohjelmistosuunnittelun vaiheet eli esitutkimus, määrittely ja suunnittelu. Käytännössä tässäkin projektissa kehitystyö oli iteratiivista eli tulosta tarkennettiin, siirryttiin seuraavaan vaiheeseen ja tarvittaessa palattiin takaisin edelliseen vaiheeseen sekä jokaisen vaiheen lopputuotokset katselmoitiin.

## **2.2 Esitutkimus**

Esitutkimuksen tarkoituksena on määritellä asiakkaan tarpeet ja asettaa yleiset järjestelmätason vaatimukset. Sen tulee vastata kysymykseen, miksi ohjelmisto tai järjestelmä tulee tehdä. Esitutkimus ei ota kantaa siihen, millainen järjestelmä täyttää asiakkaan vaatimukset. Usein esitutkimus liitetään osaksi määrittelyvaihetta, koska käytännössä asiakasvaatimusten tarkentaminen jatkuu koko määrittelyvaiheen ajan. (Haikala – Märijärvi 2006, 37.)

## **2.3 Määrittely**

Määrittelyvaiheessa analysoidaan asiakasvaatimuksia ja muutetaan ne ohjelmistovaatimuksiksi, joiden perusteella määritellään toteutettava järjestelmä. Määrittelyssä ei oteta vielä kantaa, miten järjestelmä toteutetaan. Usein myös käyttöliittymä suunnitellaan määrittelyvaiheessa. Vaatimusmäärittelyksi (requirements specification) kutsutaan dokumenttia, johon kerätään järjestelmän eri sidosryhmien vaatimukset. (Haikala – Märijärvi 2006, 37; Hiltunen.)

Toiminnallisessa määrittelyssä kuvataan kaikki järjestelmän toiminnot ja liitännät järjestelmän ulkopuolelle. Toiminnallinen määrittelydokumentti (functional specification) kuvaa sitä, mitä kaikkea järjestelmällä voi tehdä sekä miten käyttäjä voi ne tehdä. Ideaalinen määrittelydokumentti kuvaa järjestelmää siten, että missään vaiheessa (tekninen suunnittelu tai sitä seuraavat vaiheet) ei ole epäselvää, miten järjestelmän tulee toimia. (Ohjelmistotuotanto. 2012.)

## **2.4 Suunnittelu**

Suunnitteluvaihe vastaa kysymykseen, miten järjestelmä toteutetaan. Toteutus kuvataan määrittelemällä järjestelmän tekniset vaatimukset. Tekninen määrittely eli arkkitehtuurisuunnitelma kuvaa tarkasti järjestelmän teknisen arkkitehtuurin. Teknisessä määrittelydokumentissa (technical requirements tai architecture) kuvataan käytetyt ohjelmointikielet ja tietokanta- ja ohjelmistoarkkitehtuurit eli kaikkien ohjelmistokomponenttien väliset yhteydet ja rajapinnat. (Nousiainen 2009; Ohjelmistotuotanto. 2012.)

### **3 ANAMNEESIKOMPONENTTI SUUNNITTELUN LÄHTÖKOHDAT**

Opinnäytetyön tavoitteena oli suunnitella Ruotsin hammashuollon Effican Anamneesikomponentti. Opinnäytetyöhön kuului esitutkimus-, määrittely- ja suunnittelutyövaiheet eli työhön ei kuulunut varsinaista komponentin toteutusta. Jokaisen työvaiheen vaihedokumentit katselmoitiin, jonka jälkeen niihin tehtiin katselmoinnissa esiin tulleet muutokset. Opinnäytetyö toteutettiin 6.10.2011–20.4.2012.

Tieto on yksi Pohjois-Euroopan johtavista tietotekniikka-, tuotekehitys- ja konsulttipalveluja tarjoavista palveluyhtiöistä. Tiedolla on n. 18 000 työntekijää ja toimintaa lähes 30 maassa. Tiedon tärkeimmät asiakastoimialat ovat tietoliikenne, autoteollisuus, media, pankki- ja vakuutusala, metsäteollisuus, terveydenhuolto ja sosiaalitoimi, valmistava teollisuus, julkinen hallinto, energia, öljy ja kaasu, kauppa ja logistiikka. Tiedon Efficapotilastietojärjestelmä on markkinajohtaja Suomen julkisessa terveydenhuollossa. Ratkaisua käyttävät sekä perusterveydenhuollon että erikoissairaanhoidon organisaatiot. (Tieto 2 minuutissa. 2011.)

Työn tilaaja Tieto Oyj:n Healthcare & Welfare, Effic Healthcare Finland, Nordic Healthcare -yksikkö työllistää neljä henkilöä Oulussa, seitsemän henkilöä Ruotsissa ja kymmenen henkilöä Intiassa. Yksikkö vastaa mm. Ruotsin hammashuollon Effic-sovelluksesta. Ruotsin hammashuollon Efficapotilastietojärjestelmä on käytössä alle puolessa Ruotsin julkisissa hammashoitoloissa. (Lahti 2011.)

#### **3.1 Efficapotilastietojärjestelmä**

Effic kuuluu Medical Device Directiven (MDD) riskiluokkaan 1, koska Effic luokitellaan diagnosointiin tarkoitetuksi laitteeksi, jota käytetään yksin tai yhdessä lääkinnällisten laitteiden kanssa hankkimaan tietoja fysiologisten tilojen, terveydentilan ja sairauksien valvomiseksi ja hoitamiseksi. Anamneesikomponen-

tille tulee olla kaikki MDD:n vaatimat asiakirjat, jotta sille saadaan CE-merkintä. (Effica Laaturekisteri - Tuotekuvaus. 2011.)

Efficaa koskevat standardit ja vaatimukset ovat

- MDD 93/2/EEC ja 2007/47/EY
- EN 60601-1 ed 3:2005 Medical electrical equipment –General requirements for basic safety and essential performance
- EN 60601-1-6 ed2 General requirements for basic safety and essential performance – Usability
- ISO 14971 Medical devices – Application of risk management to medical devices
- EN 62304 Medical device software – Software life cycle processes (Effica Laaturekisteri - Tuotekuvaus. 2011).

Kuten useat muutkin suuret tietojärjestelmät, Effica koostuu moduuleista. Tuotteen tilaaja valitsee haluamansa moduulit ja järjestelmä rakennetaan jokaiselle tilaajalle yksilöllisesti. Tällä hetkellä Efficaa ollaan muuttamassa komponenttipohjaiseksi järjestelmäksi, jotta se saadaan vielä helpommin hallittavaksi, päivitettäväksi ja monikäyttöisemmäksi sovellukseksi.

Anamneesikomponentti on Effica-sovellukseen liitettävä erillinen komponentti, jossa näkyy potilaan esitiedot: tiedot sairauksista, lääkityksestä, elämäntavoista ja hoitotoivomuksista. Anamneesiin tuli suunnitella rakenteet, jotta anamneesiin voidaan lisätä tietoja ja tietoja voidaan muokata. Tulevaisuudessa tiedot anamneesiin voivat tulla myös potilaan täyttämältä web-lomakkeelta. Työhön kuului komponentin arkkitehtuurin, rajapintojen ja tietokantataulujen suunnittelu. Komponentti suunniteltiin asiakkaiden antamien määritelmien mukaisesti. Koska Tiedon Nordic Healthcare -yksikön dokumenttien kirjoituskielenä on englanti, kirjoitettiin opinnäytetyöhön kuuluvat dokumentit englanniksi.

Koska Anamneesikomponentin suunnitteluun vaikuttavat monet lait ja direktiivit, koin tärkeäksi työni aluksi tutustua yleisesti terveydenhuollon tietotekniikkaan,



työhön liittyviin terveydenhuollon direktiiveihin ja standardeihin sekä tärkeimpänä siihen, millainen on hammashuollon Effica.

### 3.2 Terveydenhuollon tietotekniikka

Tietotekniikkaa on hyödynnetty terveydenhuollossa jo 1960-luvulta lähtien. 1980-luvun aikana tietojärjestelmät tulivat hoitohenkilökunnan käyttöön ja 1990-luvun loppupuolelta lähtien rakennettiin potilastieto-, terveyskeskus- ja hallintojärjestelmät PC-koneisiin. Nykyaikainen potilaskansio ei koostu enää paperidokumenteista vaan tietokonepohjaisista potilastietojärjestelmistä. (Mäkelä 2006, 14–17, 35.)

Terveydenhuollon tietotekniikka voidaan jakaa neljään peruskäyttökohteeseen:

- potilasjärjestelmät: potilaan terveyteen, hoitoon ja terveydentilaan liittyvien tietojen tallennus ja käsittely
- hallintojärjestelmät: terveydenhuollon organisaatioiden hallinnollisten tietojen käsittely
- tietokonepohjaiset tutkimus- ja kuvantamisjärjestelmät: digitaalisilla kuvantamislaitteilla kuvattujen lääketieteellisten kuvien käsittely ja arkistointi
- erillisjärjestelmät: potilaiden diagnostiikka, etäseuranta, -valvonta ja -hoiva (Mäkelä 2006, 35).

Potilastietojärjestelmä yhdistää potilaaseen liittyvät tiedot muihin terveydenhuollossa käytettäviin tietoihin. Potilastietojärjestelmät voidaan jakaa kahteen osioon:

- dokumentti: kokoelma tietoa, joka arkistoidaan, esimerkiksi potilaan käyntiin liittyvä tieto hänen terveydentilastaan ja hoidosta
- viesti: tietotyyppi, joka välitetään organisaation sisällä tai organisaatioiden välillä, esimerkiksi toimenpidepyyntö tai vastaus toimenpidepyyntöön. (Mäkelä 2006, 35–38.)

Terveydenhuollon tietotekniikan keskeisimmät tekniikat ja standardit voidaan jakaa kolmeen osaan:

- yleiset tietotekniikan teknologiat eli ohjelmistoihin ja tietokoneisiin yleisesti liittyvät tekniikat ja käsitteet
- viestintä- ja tiedonsiirtoteknologiat eli tiedonsiirtoverkkoihin liittyvät käsitteet
- terveydenhuollon sovellusten standardit, erityisesti terveydenhuollon ohjelmistoihin ja tietokantoihin liittyvät käsitteet (Mäkelä 2006, 87–88).

### **3.3 Terveydenhuollon sovellusten direktiivit ja standardit**

Terveydenhuollon lääkintälaitteille ja -ohjelmistoille on tehty erilaisia standardeja potilasturvallisuuden parantamiseksi. Standardien tarkoituksena on yhdenmukaistaa teknisiä ratkaisuja ja prosesseja eri laitevalmistajien välillä. (Knuutila 2010, 2.)

Vuonna 2010 tuli voimaan lakiuudistus, joka liittyy myös terveydenhuollon ohjelmistot lain 629/2010 piiriin. Ohjelmisto on terveydenhuollon laite, kun se on lain 629/2010 5§:n kohdan 1 määritelmän mukainen: ohjelmaa käytetään mm. sairauden diagnosointiin, ehkäisyyn, tarkkailuun, hoitoon tai lievitykseen. (Knuutila 2010, 2.)

Terveydenhuollon laitteiden olennaiset vaatimukset täyttyvät silloin, kun laite on suunniteltu, valmistettu ja varustettu sitä koskevien kansallisten standardien mukaisesti. Laitteen tulee olla käyttötarkoitukseen sopiva ja sen tulee saavuttaa sille suunniteltu toimivuus ja suorituskyky. Laitteen asianmukainen käyttö ei saa aiheuttaa vaaraa potilaan, käyttäjän tai muun henkilön terveydelle tai turvallisuudelle. (L 629/2010.)

Kun terveydenhuollon laite viedään markkinoille, se merkitään CE-merkinnällä. Valmistaja osoittaa CE-merkinnällä, että terveydenhuollon laite täyttää sitä kos-

kevat olennaiset vaatimukset. Vaatimustenmukaisuus voidaan osoittaa yhdenmukaistettujen standardien avulla. (Knuutila 2010, 11.)

Ohjelmistoja arvioidaan direktiivien (MDD, IVDD) ja standardien ISO 13485, ISO 14971 JA IEC 60601-1-4 tai IEC 62304 vaatimusten pohjalta. Valmiista ohjelmistosta on vaikea osoittaa sen turvallisuutta ja vaatimustenmukaisuutta, joten ohjelmistojen turvallisuuden arviointiin liitetään

- tuottavan prosessin arviointi
- suunnitteludokumentaation arviointi. (Pöyhönen 2010, 2.)

Toisin sanoen, jotta ohjelmisto voidaan osoittaa olevan standardien mukainen, ohjelmistokehityksen jokaisen vaiheen täytyy olla dokumentoitu.

Koska työssä suunniteltava sovellus tulee Ruotsin hammashuollon Efficaan, tulee anamneesikomponentin täyttää myös Ruotsin potilastietolain SOSFS 2008:14 vaatimukset. Tätä lakia sovelletaan terveydenhuollon tarjoajiin ja se koskee henkilötietojen sähköistä käsittelyä terveydenhuollossa. Laki sisältää myös säännökset velvollisuudesta ylläpitää potilastietoja. Lain tarkoituksena on edistää potilastietojen tietoturvallista sähköistä käsittelyä ja varmistaa, että

- potilaan tiedot ovat saatavilla toimivaltaisilla terveydenhuollon tarjoajilla
- potilaan tiedot ovat virheettömiä
- asiattomat henkilöt eivät voi käyttää potilastietoja
- tietojärjestelmän käyttäjä voidaan jäljittää ja tunnistaa. (SOSFS 2008:14.)

### **3.4 Suun terveydenhuollon Effican tuotekuvaus**

Effica-sovellus on toteutettu pääosin Microsoft Visual Studio -ohjelmointiympäristöllä ja C#-ohjelmointikielellä. Sovelluksen tuotantoympäristön tietokantaohjelmisto sijaitsee SQL Serverillä. Effica-sovellukseen julkaistaan vuosittain versiopäivityksiä. (Suun terveydenhuolto - Tuotekuvaus. 2010.)

Effica sisältää suun terveydenhuollon hoitotiedot, ajanvarauksen, hoitoonkutsun ja väestörekisteritiedot. Sen käyttötarkoitus on suun terveydenhuollon potilaiden

tutkimuksen ja hoidon järjestäminen, hoidon suunnittelu ja toteutus, potilaslaskutus sekä tilastotietojen kerääminen. (Suun terveydenhuolto - Tuotekuvaus. 2010.)

Effican suun terveydenhuollon tietojärjestelmään kuuluu useita erilaisia lomakkeita ja komponentteja, jotka on yhdistetty Efficään (osa käsitellään tarkemmin luvuissa 3.3.1 ja 3.3.2), sekä erikseen ostettavia tuotteita, kuten kuvantaminen ja esilaskutus. Muita tuotteita ovat vielä oikomisanalyysi, tekstiviestimuistutus ja Kansalaisen ajanvaraus. (Suun terveydenhuolto - Tuotekuvaus. 2010.)

### **3.4.1 Yleiset lomakkeet Efficassa**

#### **Suu ja hampaisto -lomake**

Suu ja hampaisto -lomakkeelle merkitään löydökset hampaista (Suun terveydenhuolto - Tuotekuvaus. 2010).

#### **Parodontologian lomakkeet**

Efficassa on useita erilaisia parodontologian lomakkeita. Parodontologian lomakkeeseen merkitään hampaiden kiinnityskudosten löydöksiä, jotka voivat aiheuttaa korjaavia toimenpiteitä. (Suun terveydenhuolto - Tuotekuvaus. 2010.)

#### **Oikomishoidon lomakkeet**

Oikomishoidon lomakkeille kirjataan oikomishoidon aloittamisen syyt, hoitosuunnitelma, käynnit ja toimenpiteet (Suun terveydenhuolto - Tuotekuvaus. 2010).

### **3.4.2 Yleiset komponentit Efficassa**

#### **Hammaskuvantaminen**

Hammaskuvantaminen-sovelluksella voidaan tehdä kuvantamistutkimus, arkistoida kuvat ja liittää kuvalinkki potilaskertomukseen (Suun terveydenhuolto - Tuotekuvaus. 2010).

## **Päiväkirja**

Päiväkirjaan merkitään potilaan hampaiston terveydentila ja hoidot päiväkirja-muodossa (DT 4.1 Dagant Solution. 2011).

## **Hoitosuunnitelma**

Hoitosuunnitelma tehdään Suu ja hampaisto -lomakkeen löydösten perusteella (Ruotsin hammashuollon Effica. 2011).

## **Kustannusarvio**

Kustannusarviossa näkyvät hammashoidon suunnitelma ja selvitys annetusta hoidosta (Suun terveydenhuolto - Tuotekuvaus. 2010).

## **Laskutus**

Laskutuksessa näkyvät laskutustiedot suoritetuista ja hyväksytyistä toimenpiteistä (Suun terveydenhuolto - Tuotekuvaus. 2010).

### **3.4.3 Sähköinen asiointi**

#### **Tekstiviestimuistutus**

Ajanvarauksesta Efficaan integroidun Tekstiviestimuistutus-palvelun avulla voidaan potilaalle lähettää tekstiviesti (Suun terveydenhuolto - Tuotekuvaus. 2010).

#### **Kansalaisen ajanvaraus**

Kansalaisen ajanvaraus -ohjelmisto on Efficaan integroitu sähköisen asiointin sovellus. Kansalaisen ajanvaraus käsittelee potilastietoja, joten se tarvitsee vahvaa tunnistusta. (Suun terveydenhuolto - Tuotekuvaus. 2010.)

#### **Interaktiivinen esitietolomake**

Efficaan on integroitu interaktiivinen esitietolomake-sovellus. Koska sovellus käsittelee potilastietoja, se tarvitsee vahvaa tunnistusta. (Suun terveydenhuolto - Tuotekuvaus. 2010.)

## **Itseilmoittautuminen**

Efficaan on mahdollista integroida digitaalinen Itseilmoittuminen-palvelu. Potilas ilmoittautuu vastaanotolle omatoimisesti digitaalisessa ilmoittautumispisteessä. Potilaan yhteys- ja ajanvaraukset luetaan Effican potilastiedoista. (Ruotsin hammashuollon Effica. 2011.)

### **3.4.4 Tekniset vaatimukset**

Työasemissa tulee olla Internet Explorer 8 -selain, johon on asennettu Adobe Flash Player 10.x -selainlaajennus. Muisti- ja prosessoritehovaatimukseen vaikuttaa työasemalla samanaikaisesti käytössä olevien sovellusten määrä. Käyttöpalvelun toimittajan tulee tarjota varalaiteratkaisu palvelinlaitteelle. Sovelluspalvelimen tulee olla Intel Pentium Quad Core tai vastaava. Sovelluspalvelimen muistintarve riippuu käytetystä asennuskonfiguraatiosta. (Effica Laaturekisteri - Tuotekuvaus. 2011.)

## 4 ANAMNEESIKOMPONENTIN VAATIMUKSET

Työn aluksi päätettiin, etten kovin paljon tutustuisi nykyiseen anamneesiin, jotta se ei vaikuttaisi uuden suunnitteluun. Uudesta anamneesista haluttiin täysin erilainen kuin nykyinen.

Työni alkoi asiakkailta pyydetyn vaatimuslistan kääntämisellä suomeksi ja englanniksi sekä tutustumalla uusiin Dagent- ja ShortMessageBox-komponentteihin ja niiden dokumentteihin.

Käyttöliittymäsuunnittelussa sain vapaat kädet eli minulle ei annettu asiakkaiden vaatimusten lisäksi muita vaatimuksia, millainen käyttöliittymän tulisi olla. Toivomuksena kuitenkin oli, että ottaisin suunnittelussa huomioon Effican yleisen tyylin, käytettävyyden ja käytön workflow'n eli työnkulun.

Asiakasvaatimusten kääntämisen jälkeen kirjoitin vaatimusmäärittelydokumentin hahmotelman asiakasvaatimusten pohjalta. Asiakasvaatimuksia tarkennettiin koko suunnittelun ajan, jolloin kirjoitin lopullisen vaatimusmäärittelydokumentin puhtaaksi varsinaisen suunnittelutyön jälkeen.

### 4.1 Ei-toiminnalliset vaatimukset

Ei-toiminnalliset vaatimukset tulivat suoraan nykyisestä käytännöstä rakentaa Effic-sovelluksia. Uusi anamneesi tuli toteuttaa erillisenä ohjelmistokomponenttina ja sen tuli käyttää Effic-ohjelmistoalustaa. Anamneesikomponentin tuli noudattaa Effic-komponenttien vaatimuksia ja Effic-alustavaatimuksia. Anamneesin arkkitehtuurin tuli noudattaa Effican arkkitehtuurin yleistä periaatetta ja tiettyä suunnittelumallia.

Anamneesiin kirjautumisen tuli noudattaa Ruotsin potilastietolakia. Sovelluksen tuli kerätä tiedot vähintään seuraavista kohdista:

- kuka on luonut tiedot ja milloin ne on luotu
- kuka on lukenut tietoja ja milloin niitä on luettu

- kuka on päivittänyt tiedot ja milloin ne on päivitetty
- kuka on poistanut tietoja ja milloin ne on poistettu. (SOSFS 2008:14.)

Sovelluksen tuli tukea Microsoft SQL Server 2005- ja Microsoft SQL Server 2008 -tietokantapalvelimia sekä sen tuli noudattaa Effican tietokantaohjeistusta ja normalisointia.

Anamneesikomponentille tuli olla Medical Device Directiven (MDD) vaatimat asiakirjat, joita on mm. loppukäyttäjän opas, pääkäyttäjän opas, opetusmateriaali, MDD-riskienhallinta-asiakirjat ja riskianalyysi, yleiset turvallisuusvaatimusraportit sekä erilaiset käytettävyyttä tarkastelevat asiakirjat.

Vaikka minulle annettiin vapaat kädet käyttöliittymä suunnitteluun, tuli sen kuitenkin noudattaa Effica-käyttöliittymäohjeistusta ja Life Care -ohjeistusta siltä osin, kuin se oli mahdollista.

#### **4.2 Toiminnalliset vaatimukset**

Myös osa anamneesikomponentin toiminnallisista määryksistä oli samoja kuin aikaisemmissa Effica-sovelluksissa. Toiminnallisia vaatimuksia myös tarkennettiin koko projektin ajan yhteistyössä asiakkaan edustajan kanssa.

Toteutuksen pääperiaatteena tuli olla, että yhdellä potilaalla on vain yksi anamneesinäkö, jossa näkyy potilaan usin anamneesi. Potilaalle voidaan luoda uusi anamneesi, jolloin edelliset anamneesit näkyvät historianäkymässä.

Asiakkaiden vaatimuksina uudelle anamneesille oli, että siinä näkyisivät tietyt potilaan tiedot. Erityisen tärkeää asiakkaille oli, että uudessa anamneesissa olisi käytössä NPÖ Blomma ja tiedon välittäminen sekä vastaanottaminen Nationell Patientöversiktillä (NPÖ).

Anamneesin käyttöliittymässä tuli näkyä kysymyksiä hoitotoiveista, sairauksista, yleisestä terveydentilasta, hampaiden hoidosta jne., joissa vastausten tuli olla



kyllä-ei-vastauksia sekä tuli olla mahdollisuus kirjoittaa vapaata tekstiä esimerkiksi hoitotoivomuksiin.

Koska asiakkaat halusivat itse määritellä, mitä kysymyksiä anamneesissa on, ohjelmiston Effican Hallinta-sovellukseen tuli tehdä rakenne, jota kautta organisaation pääkäyttäjä pystyy määrittelemään, mitä kysymyksiä anamneesin käyttöliittymässä näkyy ja millainen kysymysten rakenne on. Effican Hallinta-sovellus on Efficassa oleva erillinen sovellus, jota kautta organisaation pääkäyttäjä voi määritellä Effic-sovellusten perusasetuksia.

Käyttäjällä tulee olla mahdollisuus sähköisesti allekirjoittaa päiväysrivi ja siihen kuuluvat tietorivit. Allekirjoittanut käyttäjä tulee näkyä päiväysrivillä allekirjoituksen jälkeen. Käyttäjällä tulee myös olla mahdollisuus allekirjoittaa toisen puolesta ja mahdollisuus mitätöidä allekirjoitetut tiedot. Kun anamneesi allekirjoitetaan, siitä tulee muodostua automaattinen päiväkirjamerkintä Daganttiin. Ruotsin potilastietolain mukaisesti allekirjoittamattomat tiedot tulee lukita automaattisesti viimeistään 14 päivän kuluttua tietojen luomisesta (SOSFS 2008:14).

### **4.3 Nationell Patientöversikt**

Euroopan Unioni asetti 31.12.2005 direktiivin kaikille jäsenvaltioilleen määritellä menettelytavat, joilla kehitetään terveydenhuollon tietotekniikkaa. Keväällä 2006 tietotekniikkastrategia oli valmis. (Berg 2009.)

Strategian erityistavoitteena on yhtenäisen sähköisen potilastietojärjestelmän luominen tukemalla tietojen vaihtoa ja standardointia. Sähköisten terveydenhuoltopalveluiden kehittämistyö on painottunut potilaiden hoitoon liittyvän tiedon sähköiseen välittämiseen hoitoon osallistuvien kesken. (Salmivalli 2010.)

Vuonna 2008 Euroopan komissio antoi suosituksen sähköisten terveystietojärjestelmien rajat ylittävästä yhteentoimivuudesta. Suosituksen tarkoituksena on kehittää yleiseurooppalaista sähköisen terveydenhuollon yhteentoimivuutta vuoden 2015 loppuun mennessä. (Salmivalli 2010.)

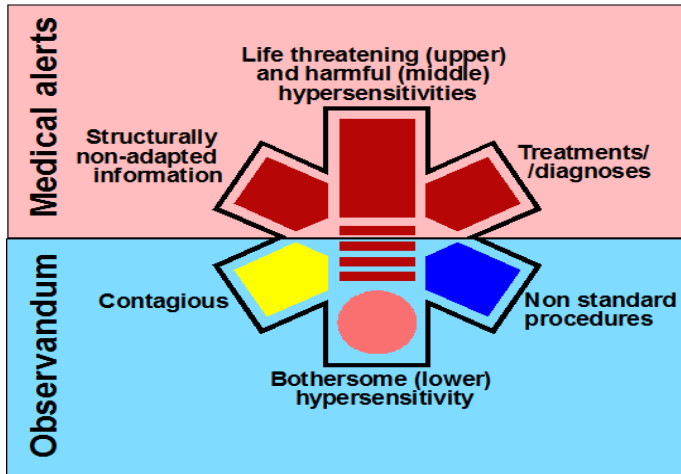
Vuonna 2009 Ruotsissa kaikki kunnat päättivät ottaa kansallisen tietotekniikkastrategian käyttöön. Kansalliseen tietotekniikkahankkeeseen kuuluu kuusi toiminta-aluetta, joista yksi on Kansallinen potilasyhteenvetotietojen eli Nationell Patientöversiktin (NPÖ:n) kehittäminen ja toinen on organisaatorajat ylittävän tiedonsiirron mahdollistaminen. (Salmivalli 2010.)

NPÖ mahdollistaa se, että potilaan suostumuksella hoitohenkilökunta tallentaa potilastietoja kansalliseen potilastietojärjestelmään, josta ne ovat käytössä useilla terveydenhuollon palvelujen tarjoajilla. Kun hoitohenkilökunnalla on käytössä kokonaiskuva potilaan terveystiedoista, heillä paremmat mahdollisuudet tehdä oikeat diagnoosit ja antaa oikeanlaista hoitoa. (Health informatics. 2009.)

Tällä hetkellä Ruotsissa on käytössä useanlaisia varoitustapoja hengenvaarallista yliherkkyyksistä, sairauksista ja tarttuvista taudeista, mutta ne ovat epäyhtenäisiä ja joskus epäselviä, mikä estää oikeantyyppisen tietojen välityksen organisaatioiden välillä. Kriittiseen tietoon pääsyn helpottamiseksi on terminologiaa ja tietorakenteita yhtenäistetty. Kriittisten tietojen osoittamiseksi kehiteltiin myös visuaalinen varoitussymboli (Alert Symbol) eli tuttavallisemmin NPÖ Blomma, joka on saanut tunnustusta monissa maissa. (Health informatics. 2009.)

#### **4.4 Alert Symbol eli NPÖ Blomma**









Visuaalisen NPÖ Blomma -kuvakkeen on tarkoitus ilmaista useita ominaisuuksia yhdellä kertaa. Siitä näkee heti, onko potilaalla hengenvaarallisia yliherkkyyksiä tai tarttuvia tauteja. Symboli on saanut vaikutteita Star of life -symbolista. Yläosa symbolista viestittää, jos potilaalla on vakavia, hengenvaarallisia tai haitallisia hoitoon vaikuttavia riskejä (kuva 3). Alaosa symbolista viestittää vähemmän vakavista, mutta silti mahdollisesti hoitoon vaikuttavista asioista. Väreistä punainen tarkoittaa yliherkkyyttä tai muuta tärkeää tietoa, joka tulee ottaa huomioon hoidossa. Tarttuvat taudit ilmaistaan keltaisella värillä, mikä yleisesti symboloi biologista vaaraa. Sininen väri ilmaisee huomioon otettavaa, mutta ei vakavaa hälytystä. (Health informatics 2009.)



© Swedish Association of Local Authorities and Regions

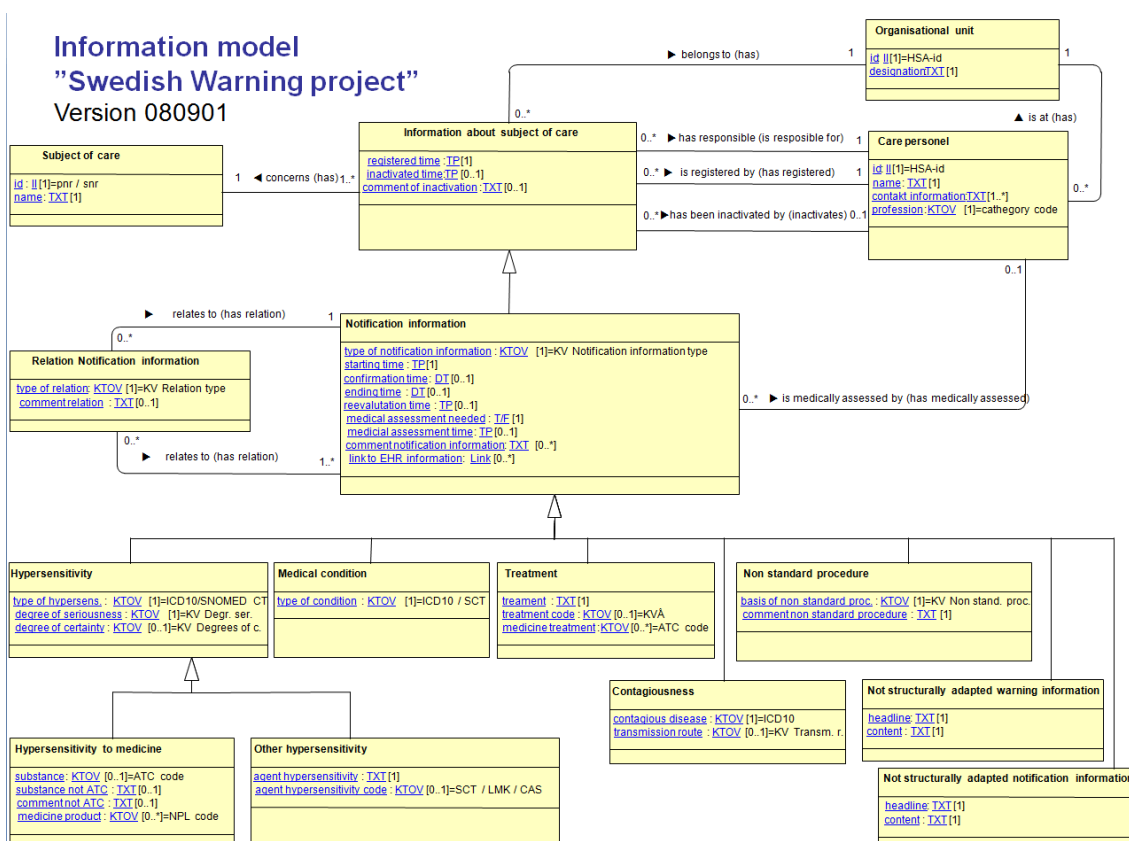
KUVA 3. NPÖ Blomma (Lövström 2009.)

Kuva 4 kertoo symbolin eri osien merkityksen. Yliherkkyyksien vakavuus näkyy symbolin keskiosassa, joka voidaan jakaa kolmeen osaan: kiusallinen, haitallinen ja hengenvaarallinen. Hengenvaarallisessa yliherkkyydessä symbolin keskelle muodostuu punainen huutomerkki. Yliherkkyyksistä tulee myös arvioida, kuinka varmaa tieto on. Huomioon otettavat sairaudet ja hoidot näkyvät symbolin oikeassa yläkulmassa punaisella. Niitä ovat mm. diabetes, astma, syöpä ja sydän- ja verisuonisairaudet. Kun potilaalla on mahdollisesti hoidossa huomioon otettava poikkeustieto, se näkyy symbolin oikeassa alakulmassa sinisellä. Näitä on esimerkiksi salainen puhelinnumero. Vasemman alakulman keltainen väri kertoo, että potilaalla on tarttuva tauti. Tarttuvista taudeista myös taudin tarttumistapa tulee tietää. Vasemman yläkulman sakara on punainen, jos potilaalla on ei-luokiteltua tietoa, joka tulee ottaa huomioon potilasta hoidettaessa. Tällaista tietoa voi olla tieto metalliosista potilaan kehossa. Symbolia käytettäessä olisi hyvä, jos symbolia klikattaessa voisi päästä tarkastelemaan tarkemmin tietoja, jotka vaikuttavat potilaan symboliin. (Health informatics. 2009.)

Uppmärksamhetssignal	Aktiverad yta i symbolen
Ingen eller alla är inaktuella	
Överkänslighet, allvarlighetsgrad = besvärande	
Överkänslighet, allvarlighetsgrad = skadande	
Överkänslighet, allvarlighetsgrad = livshotande	
Sjukdom/behandling	
Vårdbegränsning	
Smitta	
Ej strukturanpassad	

KUVA 4. Symbolin osien merkitys (Fagerberg 2011)

Kuvassa 5 näkyy tietomalli varoitujärjestelmästä. Mallissa esitettävät tiedot on koottu luokkiin ja malli esittää näiden luokkien väliset suhteet. (Lövström 2009.)



KUVA 5. Tietomalli varoitujärjestelmästä (Lövström 2009)

## 4.5 Käytettävyys

Käyttöliittymällä tarkoitetaan laitteen, ohjelmiston tai minkä tahansa muun tuotteen osaa, jonka avulla käyttäjä käyttää tuotetta. Käyttöliittymän voidaan katsoa olevan kokonaisuus, joka koostuu tuotteen viesteistä ja käytännöllisistä osatuotteista, esimerkiksi valikoista, painikkeista ja näytöstä. Ohjelmistotuotteen käytön osaaminen pohjautuu käyttöliittymän havaitsemiseen ja ymmärtämiseen. Käyttäjä arvioi sovelluksen laatua lähes yksinomaan käyttöliittymän perusteella. (Käyttöliittymä. 2012)

Se, että tuotetta on helppo käyttää, ei vielä tee tuotteen käytettävyydestä hyvää. Käytettävyys on tuotteen laatuominaisuus, joka kertoo, kuinka tehokasta ja helppoa tuotetta on käyttää. Kansainvälisen standardisoimisjärjestön ISO:n 9241–11-standardi määrittelee käytettävyyden mittariksi, jolla mitataan tuotteen käytön tuottavuutta, tehokkuutta ja miellyttävyyttä. Jacob Nielsen taas määrittelee käytettävyydelle viisi laatukomponenttia: opittavuus, tehokkuus, muistettavuus, virheettömyys ja miellyttävyys. On myös joukko muita käytettävyyteen liitettyjä ominaisuuksia, jotka kaikki määrittelevät tuotteen ominaisuuksia. Eri käytettävyysohjeet ovat nyrkkisääntöjä, joita tuotteen suunnittelussa tulisi noudattaa. Ohjeissa on tietoa suunnittelijoille, joilla ei ole mahdollista perehtyä ihmisen ominaisuuksiin tuotteen käyttäjänä. (Sinkkonen 2004.)

Hyvä käyttöliittymä opastaa käyttäjää oikeaan, virheettömään käyttöön ja sen käyttö on helppo oppia. Käyttöliittymän käyttötavasta tulee ”itsestään selvää”. Hyvän järjestelmän tuotannon laatu on korkeaa, se on nopea konfiguroida ja keskeytyksiä tapahtuu vähän. Sen käyttö on turvallista, koska virhetilanteita ei synny ergonomisten puutteiden tai huonon logiikan johdosta tai, jos niitä tulee, virhetilanteet hallitaan siten, ettei niistä seuraa mitään pahempaa. Hyvän käyttöliittymän työnkulku on automaattista, siinä tehtäväosiot seuraavat johdonmukaisesti toisiaan ja käyttäjä ei pääse helposti harhailemaan päätehtäväpolulta. Käyttöliittymän laadun ratkaisee lopulta se, miten hyvin se sopii käyttäjälle, käyttötehtävien kokonaisuuteen ja käyttöolosuhteisiin. (Vuori – Kivistö-Rahnasto - Toivonen 1998.)

Käyttöliittymissä on suositeltavaa käyttää elementtejä, jotka ovat ennestään tuttuja muista ohjelmista. Käyttöliittymä elementtejä ovat mm. painonapit, valikot, vierityspalkit ja kuvakkeet. Eri käyttöliittymien samankaltaisuus auttaa käyttäjää oppimaan nopeasti uuden ohjelman toiminnan. Käyttöliittymän visuaalissa suunnittelussa on hyvä pyrkiä loogiseen ja yhtenäiseen ulkoasuun. Käyttöliittymän näkymä kannattaa jakaa osiin siten, että tietyt peruselementit pysyvät loogisesti samoissa paikoissa läpi sovelluksen, vaikka siirryttäisiin ikkunasta toiseen. Suunnittelussa olisi hyvä pyrkiä muuttumattomuuteen, jotta käyttäjän on helppo omaksua näkymä heti sille tullessaan. (Rouhiainen 1997.)

## 5 ANAMNEESIKOMPONENTIN TOIMINNALLISET RATKAISUT

Eri oppaissa ohjelmistotuotannossa vaatimusmäärittelyjen jälkeen tehdään toiminnallinen määrittely, jonka jälkeen aletaan suunnitella käyttöliittymää ja ohjelmiston arkkitehtuuria. Tämän projektin kuitenkin aloitin luonnostelemalla ja suunnittelemalla käyttöliittymää. Käytin Powerpointia käyttöliittymäsuunnittelun työvälineenä. Suunnittelun aluksi luonnostelin erilaisia käyttöliittymäehdotelmia. Luonnoksista valitsimme työpaikkaohjaajani kanssa sen luonnoksen, jonka suunnittelua jatkaisin. Käyttöliittymäsuunnittelun aikana suunnittelin myös erilaisia ratkaisuja toteuttamaan anamneesikomponentin vaaditut toiminnot.

Käyttöliittymäsuunnitelmassa esitin Powerpointin eri dioissa käyttöliittymän toimintaa. Käyttöliittymän ulkoisen olemuksen, työkalut ja toiminnot pyrin pitämään samantyyppisinä kuin ne ovat uudessa Dagentissa ja muissa uusimmissa Efficakomponenteissa. Anamneesikomponentin toteutusvaiheessa tulee anamneesin käyttöliittymän otsikoinnit, väritys ja työkalut muuttaa uusimpien ohjeistuksien mukaiseksi.

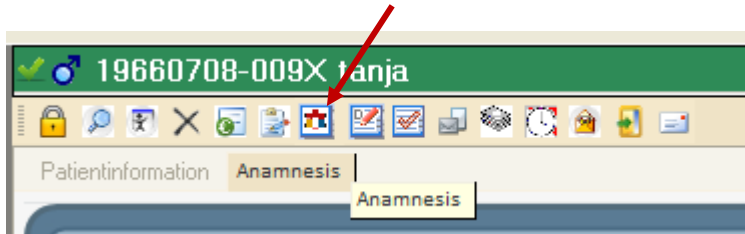
Kun käyttöliittymän ulkoasu ja toiminnot olivat pääpiirteissään selvillä, aloitin varsinaisen toiminnallisen määrittelydokumentin kirjoittamisen. Toiminnallisen määrittelyn aikana tarkennettiin asiakasvaatimuksia yhteistyössä asiakkaan edustajien kanssa eri kohtien halutusta rakenteesta ja toiminnoista.

Tämän projektin toiminnallisessa määrittelydokumentissa kuvataan anamneesikomponentin toimintaa ja sen eri näkymien, näyttöliittymäkontrollien ja asetusten toimintaa.

*Seuraavaksi on kuvattu anamneesikomponentin toiminnallisia ratkaisuja. Koska ne ovat salaista tietoa, on osa tiedoista poistettu sekä kuvia muokattu alkuperäisestä.*

## 5.1 Anamneesin avaaminen

Anamneesi voidaan avata navigaatiolistalta tai valikkopalkista sitä varten olevasta ikonista (kuva 6).



KUVA 6. Anamneesi-ikoni valikkopalkissa

Kun anamneesi avataan navigaatiolistalta tai valikkopalkin ikonista, anamneesinäkymä avautuu. Anamneesin rakenne on seuraavanlainen (kuva 7):

Anamnesis			
2011-11-26 tdl Dentist Tina det Test Clinic 001			
<b>Care wish</b>			
Anesthesia	Hopes that a lidocaine is being used		
<b>Diseases</b>			
Hemopathy, anaemia	Note	Medicine	Dosage
	Some text written on th		
Pulmonary disease, Asthma		Bulmicort Inhaler 200 µg	1 inhalation twice day
<b>General health</b>		<b>Note</b>	
Nut allergy			
Cleans the teeth twice a day and use dental floss			

KUVA 7. Anamneesinäkymä

Anamneesissa käytettävissä olevat koko ajan näkyvät työkalut ovat (kuva 8) Uusi Anamneesi, Tallenna, Tulosta, Sanahaku, Lista liitteistä, Historia, Anamneesi ja NPÖ Blomma. Työkaluista kaikki muut ovat samanlaisia ja toimivat sa-



malla tavalla kuin Dagantissa paitsi Uusi Anamneesi-, Historia- ja NPÖ Blomma -työkalut.



*KUVA 8. Anamneesissa koko ajan näkyvät työkalut*

*Seuraavaksi on kuvattu tarkemmin anamneesin rakennetta. Koska se on salais-  
ta tietoa, se on poistettu tästä julkisesta versiosta.*

## **5.2 Uuden anamneesin luominen**






*Seuraavaksi on kuvattu tarkemmin uuden anamneesin luominen. Koska se on  
salaista tietoa, se on poistettu tästä julkisesta versiosta.*

## **5.3 Anamneesin tietojen muokkaaminen**

*Seuraavaksi on kuvattu tarkemmin anamneesin tietojen muokkaaminen. Koska  
se on salaista tietoa, se on poistettu tästä julkisesta versiosta.*

## **5.4 NPÖ Blomma**

NPÖ Blommassa näkyvät potilaan yliherkkyydet, niiden vakavuus ja luotetta-  
vuus, sairaudet, tarttuvat taudit sekä muut potilaan hoitoon vaikuttavat tiedot.  
NPÖ Blomma -näkömään pääsee valitsemalla oikean yläkulman NPÖ Blomma  
-työkalun, jolloin näkymä avautuu (kuva 9).

Anamnesis					
2011-11-26 <a href="#">View Details</a> <a href="#">Test Cases</a> 001					
	<u>Hypersensitivities</u>	<u>Degree of seriousness</u>	<u>Degree of safety</u>	<u>Date</u>	<u>Actuality</u>
	<a href="#">Peanuts</a>	<a href="#">Life-threatening</a>	<a href="#">Suspected</a>	2009-11-01	
	<u>Structurally non-adapted information</u>	<u>Contents</u>	<u>Date</u>	<u>Actuality</u>	
	<u>Treatments/Diagnoses</u>		<u>Date</u>	<u>Actuality</u>	
	<a href="#">Pulmonary disease, Asthma</a>		2009-11-01		
	<u>Contagious</u>	<u>Transmission route</u>	<u>Date</u>	<u>Actuality</u>	
	<u>Non standard procedures</u>		<u>Date</u>	<u>Actuality</u>	

KUVA 9. NPÖ Blomma -näkyvä

## 5.5 Tietorivien kysymysten ja otsikoiden lisääminen

Anamneesin näkymässä näkyvät kysymykset ja kysymysrakenteet ovat kaikki organisaatioiden itsensä määrittämiä, ainoastaan otsikkoriveillä näkyvät otsikot ovat pysyviä. Effican Hallinta-sovellukseen tulee lisätä uusi anamneesinäkyvä (kuva 10), jossa organisaation pääkäyttäjällä voi määrittellä, mitä kysymyksiä anamneesissa näkyy ja millainen kysymysten rakenne on.

*Seuraavaksi on kuvattu tarkemmin Hallinta-sovelluksen anamneesinäkyvän toimintaa. Koska se on salaista tietoa, se on poistettu tästä julkisesta versiosta.*

## 5.6 Anamneesi sähköiseltä esitietolomakkeelta

*Seuraavaksi on kuvattu tarkemmin anamneesi, joka tulee sähköiseltä esitietolomakkeelta. Koska se on salaista tietoa, se on poistettu tästä julkisesta versiosta.*

## 5.7 Historianäkymä

Anamneesin historiatiedot näkyvät historianäkymässä (kuva 36). Historianäkymään pääsee valitsemalla työkaluriviltä Historia-työkalun.

*Seuraavaksi on kuvattu tarkemmin historianäkymän rakennetta. Koska se on salaista tietoa, se on poistettu tästä julkisesta versiosta.*

## 6 ANAMNEESIKOMPONENTIN ARKKITEHTUURISUUNNITELMA

Arkkitehtuurisuunnittelun aloitin tutustumalla suunnitteilla olevan Dental Status -komponentin arkkitehtuuriin, koska anamneesin arkkitehtuuri tuli toteuttaa vastaavalla tavalla. Anamneesin arkkitehtuurirakenteesta tuli käytännössä vastaavanlainen kuin on Dental Status -komponentin arkkitehtuuri. Anamneesin arkkitehtuurin suunnittelussa lähinnä tuli miettiä, tuleeko anamneesista olla rajapintoja muihin ulkopuolisiin sovelluksiin. Ruotsin hammashuollon Efficia on integroitu Dental Frameworkiin, joka perustuu Open Frameworkiin. Anamneesin toteutuksessa ei tule kuitenkaan käyttää moduuleita, jotka perustuisivat Open Frameworkiin. Anamneesikomponentin arkkitehtuurin tuli rakentua tietyn suunnittelumallin mukaisesti ja sen tuli pohjautua Windows-ratkaisuihin.

### 6.1 Komponenttipohjainen ohjelmistokehitys

Erilaiset sovellukset, niin ohjelmistot kuten laitteetkin voivat rakentua itsenäisesti, erikseen kehitettävistä ja yhteen koottavista komponenteista, joten ne ovat laajennettavissa ja ylläpidettävissä komponentteittain. Komponenttipohjaisessa ohjelmistotyössä komponenttien kehittäminen on erillään niiden hyödyntämisestä sovelluksissa. (Kuikka 2012.)

Ohjelmistokomponenttien avulla on mahdollista saada entistä laadukkaampia ja paremmin testattuja järjestelmiä. Komponenttipohjainen ohjelmistokehitys perustuu rajapintakeskeiseen suunnitteluun, jossa komponenttien tarjoamia palveluita voidaan myös testata rajapintojen avulla ulkoapäin. Komponenttien kehittämisessä voidaan myös hyödyntää yleisiä sekä toimialakohtaisia standardeja. (Kuikka 2012.)

Komponenttiperustaisessa ohjelmistokehityksessä komponenttimalli määrittelee standardit ja menettelytavat, joita komponenttien kehittäjien ja hyväksikäyttäjien tulee noudattaa. Komponenttimalli ohjeistaa, miten yksittäinen komponentti kehitetään ja miten komponentteja otetaan käyttöön rajapintoja hyväksikäyttäen. Sovelluksissa komponentit kootaan liittämällä niitä komponenttialustaan,

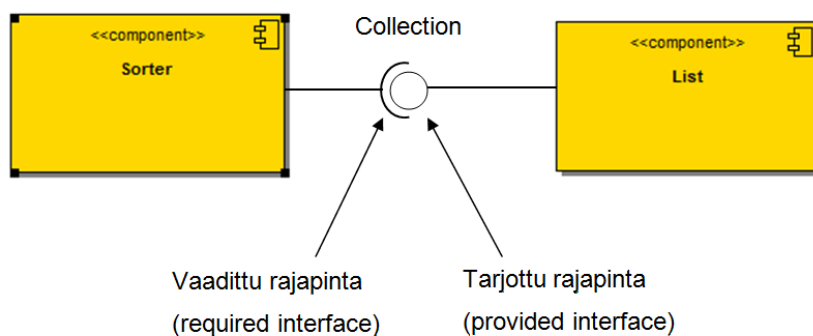
-kehykseen tai yhteen toisten komponenttien kanssa komponenttirajapintojen avulla. Komponenttialusta saa kehittäjän toimimaan mallin ehdoilla sekä tukee mallia palvelujen ja suoritusympäristön avulla. (Kuikka 2012.)

### **6.1.1 Komponentit ja rajapinnat**

Ohjelmistokomponentti on itsenäinen ohjelmistoyksikkö, joka tarjoaa palvelujaan hyvin määriteltyjen rajapintojen kautta. Komponentti on myös arkkitehtuurielementti, joka sisältää osan järjestelmän toiminnallisuutta, joka rajoittaa pääsyn tähän osaan määriteltyjen rajapintojen kautta ja jolla on määritellyt riippuvuudet suoritusympäristöstään. Muista ohjelmayksiköistä riippumatta komponenttia voidaan käyttää yhtenä yksikkönä olettaen, että rajapinnat on toteutettu ympäristön vaatimusten mukaisesti. (Koskimies – Mikkonen 2005, 53–54; Laine 2011.)

Komponentin koolla ei ole rajoituksia. Se voi olla yksinkertainen palveluja tarjoava olio tai se voi olla myös useita luokkia käsittävä sovellus. Komponentti muodostaa oman toiminnallisen kokonaisuuden. Komponenteilla saadaan järjestelmä muunneltavaksi. Järjestelmässä voidaan lisätä, vaihtaa ja poistaa komponentteja sen mukaan, millainen toiminnallisuus järjestelmään halutaan. Jotta komponentti on uudelleen käytettävä, sen tulee noudattaa jonkin komponenttimallin asettamia vaatimuksia. (Koskimies – Mikkonen 2005, 54–57; Kuikka 2012.)

Rajapinnat on keskeinen osa ohjelmistoarkkitehtuuria. Suurin osa arkkitehtuurityyleistä ja -malleista perustuu rajapintojen käyttämiselle. Rajapinnat määrittävät tavat, joilla komponentit keskustelevat keskenään. Rajapintojen huolellinen suunnittelu mahdollistaa kehitystyön järkevän jakamisen, testattavuuden, ylläpidettävyyden ja joustavuuden. Komponentti voi tarjota rajapinnan ja se voi vaatia rajapinnan mukaisia palveluja eli yksi rajapinta voi olla yhden komponentin kannalta vaadittu ja toisen kannalta tarjottu (kuva 10). (Koskimies – Mikkonen 2005, 57–71; Laine 2011.)



KUVA 10. Tarjotut ja vaaditut rajapinnat (Laine 2011)

### 6.1.2 Ohjelmistoarkkitehtuuri

Ohjelmistoarkkitehtuuri on ohjelmiston kehittämisen ydin. Arkkitehtuurin tehtävänä on antaa yleiskuva järjestelmän toiminnasta ja ei-toiminnallisista ratkaisuista, ottaa kantaa ohjelmiston olennaisiin ratkaisuihin sekä kuvata ohjelmiston jako pääosiin, osien väliset suhteet, kommunikointi ja sijoittelu. Arkkitehtuuri määrittelee myös muut ohjelmistoon ja sen kehittämiseen liittyvät säännöt. Arkkitehtuurikuvauksen tulee varmistaa vaatimusten toteutuminen, joten sen tulee olla kattava, ristiriidaton ja yksiselitteinen. (Laine 2011.)

Ohjelmistoarkkitehtuurille on olemassa useita erilaisia määritelmiä. IEEE:n standardi määrittelee ohjelmistoarkkitehtuurin järjestelmän perusorganisaatioksi. Tämä standardi määrittelee järjestelmän osat, niiden keskinäiset suhteet ja suhteet ympäristöön sekä periaatteet, jotka ohjaavat järjestelmän kehittymistä. (Koskimies – Mikkonen 2005, 18.)

Ohjelmistoarkkitehtuuri on järjestelmän kartta ja opas, joka antaa rajat ja helpottaa järjestelmän rakentamista, testausta, ylläpitoa ja uudelleenkäyttöä. Ohjelmiston jako osiin auttaa hallitsemaan kokonaisuutta, mahdollistaa valmiiden osien hyväksikäytön ja hajautuksen, helpottaa ratkaisun ymmärtämisessä ja mahdollistaa työnjaon toteutuksessa. Hyvä ohjelmistoarkkitehtuuri tukee myös ohjelmiston laatuominaisuuksia. (Laine 2011; Koskimies – Mikkonen 2005, 19.)

Yksi ohjelmistotuotannon merkittävimmistä haasteista on ohjelmistojen uudelleenkäyttö: miten samoja osia voidaan hyödyntää useissa eri ohjelmistotuotteissa. Ohjelmistojen uudelleenkäyttö pohjautuu aina arkkitehtuuritason ratkaisuihin. Parhaimmillaan tuotekategorialla on yhteinen ohjelmistoarkkitehtuuri ja sillä on sitä tukeva ohjelmistoalusta, jonka päälle on toteutettu yksittäiset tuotteet. (Koskimies – Mikkonen 2005, 157–159.)

Komponenttiarkkitehtuurin suunnittelussa suunnittelumallien ja sovelluskehysten käyttö on keskeistä. Hyvä komponenttipohjainen arkkitehtuuri on skaalautuva, tehokas, selkeä ja ymmärrettävä sekä se sallii riippumattomien komponenttien itsenäisen kehityksen. Suunnittelussa on erityisesti otettava huomioon komponenttien ja sovelluskehysten vuorovaikutukset sekä komponenttien roolit ja koontimahdollisuudet. Hyvä suunnittelumalli auttaa toteuttamaan ja ylläpitämään käyttöliittymää sen toteuttaessa käyttäjien vaatimuksia ja liiketoimintäsääntöjä. (Kuikka 2012.)

### **6.1.3 Model View ViewModel -arkkitehtuurimalli**

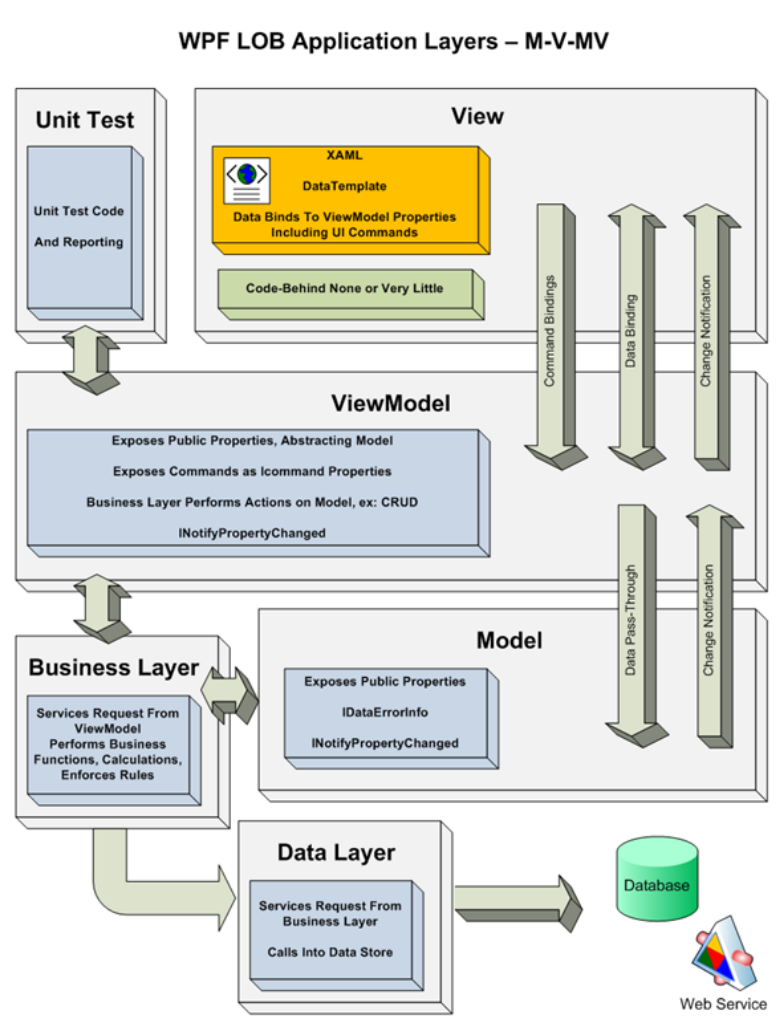
Vuonna 2006 julkaistu Model View ViewModel (MVVM) -arkkitehtuurimalli suunniteltiin tukemaan WPF:ää ja Silverlightia. MVVM ja WPF auttavat ylläpidettävien ja mukautuvien käyttöliittymien kehittämisessä. MVVM perustuu Model-View-Controller-arkkitehtuurimalliin (MVC), jonka tarkoituksena on ollut käyttöliittymän erottaminen sovellusalueiedoista. (Model View ViewModel. 2012; MVC-arkkitehtuuri. 2011.)

Molemmat arkkitehtuurimallit sisältävät kolmenlaisia komponentteja: malli-, näkymä- ja ohjainkomponentit. Malli- eli Model-komponentit sisältävät sovelluksen tietorakenteet ja ovat vastuussa kaikesta logiikasta, tiedon noutamisesta ja tallentamisesta. Näkymä- eli View-komponentit määrittävät käyttöliittymän ulkoasun ja vastaavat tiedon esittämisestä käyttäjälle. ViewModel, joka vastaa MVC:n Controlleria eli ohjainta, hoitaa tiedon välittämisen mallista näkymälle siinä muodossa kuin sitä tarvitaan. ModelView sisältää käyttöliittymän taustalogiikan ja toiminnot luodaan metodeina, joita näkymä kutsuu. ViewModelin ero

MVC:n Controlleriin on, ettei ViewModel tunne näkymäosiota. Kuvassa 11 näkyy MVVM-arkkitehtuuri. (MVC-arkkitehtuuri. 2011; Kallonen 2011.)

MVVM-mallin idea on siirtää koko käyttöliittymän toiminta erilleen sovelluksen toiminnasta. Mallin etuja ovat mm. seuraavat:

- ViewModelin testattavuus eli ViewModelia testataan käyttämällä standardeoituja yksikkötestejä
- mahdollisuus suunnitella käyttöliittymä uudelleen, koska rajapinta View-Modeliin pysyy samana
- mallin uudelleenkäytettävyys
- kehitys- ja ylläpitokustannusten pienentäminen. (Mosers 2011.)



KUVA 11. MVVM-arkkitehtuuri (Shifflett 2008)



Käyttöliittymä toteutetaan käyttämällä Extensible Application Markup Language (XAML) ja toteutuksessa tulee pyrkiä välttämään code-behind-koodia. MVVM käyttää WPF-mallia, jossa näkymään syötetyt käyttäjän komennot välittyvät databindingillä ViewModeliin (kuva 11). ViewModel osioon toteutetaan INotifyPropertyChanged-rajapinta, joka mahdollistaa näkymän reagoimisen tiedon muutoksiin. (Kallonen 2011.)

Arkkitehtuuria toteutettaessa tulee välttää eventtejä ja niiden tilalla käytetään Commandingia. Käyttöliittymän toiminnallisuus liitetään ViewModel osioon Commandingilla. View sidotaan ViewModeliin luomalla ilmentymä ViewModelista, jonka se sitoo itseensä DataContextilla tai XAML:llä. (Kallonen 2011.)

## **6.2 Käytettävät teknologiat**

### **6.2.1 Open Framework**

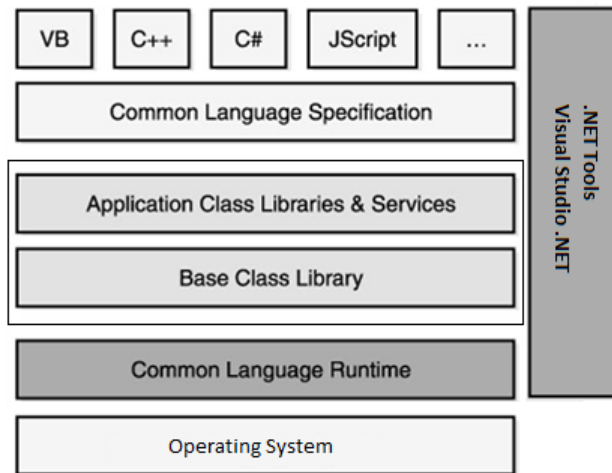
Open Framework on luokkakirjasto, joka suunniteltiin luovaan koodaukseen. Open Framework tarjoaa erilaisia toimintoja, kuten sisäänkirjautumisen yhteydessä jakaa tietoja käyttäjän ja valitun potilaan kanssa, sekä Plug and Play -tekniikan, jolla sovellus voidaan helposti yhdistää osaksi muita sovelluksia vain muuttamalla asetuksia. (Dental Architecture Guideline. 2010.)

### **6.2.2 .NET Framework**

.NET Framework on Microsoftin kehittämä Visual Studio .NET -ympäristössä käytettävä ohjelmistokomponenttikirjasto. C# ja Visual Basic ovat käytetyimmät ohjelmointikielet noin 20 tuetusta ohjelmointikielestä. Käyttämällä .NET Frameworkia voidaan ohjelmistoja kehittää suorituskykyisiksi, helposti laajennettaviksi ja tietoturvalisiksi vähäisellä ohjelmakoodimäärällä. (.NET Framework. 2011.)

Microsoftin kehittämä .NET Platform on ohjelmistojen kehitysalusta, joka luo rungon alustan päälle rakennettavalle sovellukselle. .NET Platform tarjoaa työkalut ja teknologiat, jotka helpottavat ja nopeuttavat ohjelmistojen toteuttamista. .NET Platform -arkkitehtuuri (kuva 12) sisältää viisi osaa: Common Language Specification, Framework Class Library (FCL), Common Language Runtime

(CLR), Operating System ja .NET Tools. (.NET Framework. 2011; Rahikainen 2011, 3–6.)



KUVA 12. .NET Framework arkkitehtuuri (Rahikainen 2011)

Ohjelmointikehityksen perustana on .NET Framework -komponentti, joista FCL ja CLR ovat kaksi keskeisintä osaa. FCL on .NET luokkakirjasto, joka sisältää tuhansia luokkia, joiden avulla sovelluksessa saadaan toteutettua erilaisia toimintoja. CLR on ohjelmien ajoympäristö, joka on Microsoftin toteutus Common Language Infrastructuresta (CLI). CLR kääntää ja tulkaa ohjelman suorituksen aikana kehitysympäristön tuottaman CIL-kielen tietokoneen ymmärtämään binäärimuotoon. (Rahikainen 2011, 3–6.)

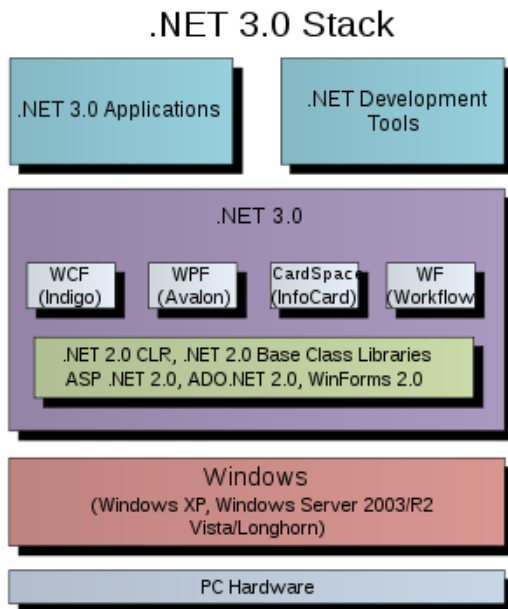
### 6.2.3 C#

Microsoftin kehittämä ja ISO-standardoitu C# on .NET Platformin pääasiallinen ja käytetyin Windows ohjelmointikieli. C#:lla voi tehdä Windows, konsoli- ja nettisovelluksia. C# on yksinkertainen, olio-orientoitunut, tyyppiturvallinen korkeamman tason ohjelmointikieli ja samalla ensimmäinen komponenttisuuntautunut kieli. (Rahikainen 2011, 3–6; Räisänen, 2011.)

### 6.2.4 WPF

.NET Framework 3.0:ssa esiteltiin käyttöliittymäkirjasto Windows Presentation Foundation (WPF), joka muodostaa graafisen rajapinnan Windows Vistan ja

myöhempien Windows-versioiden kanssa (kuva 13). WPF on vektoripohjainen ja resoluutioriippumaton sekä se sisältää työkalut, joilla voi luoda laitteistokiihdytystä, mediasisältöä ja modernimpia käyttöliittymäominaisuuksia kuin Windows Forms. (Windows Presentation Foundation. 2011; Ekonoja – Lahtonen 2012.)



KUVA 13. WPF osana .NET Frameworkia (Windows Presentation Foundation. 2011)

WPF toimii käyttöliittymän pohjana, jossa ulkoasu voidaan erottaa koodista ja sen kuvaamiseen voidaan käyttää erillisiä tyylimäärittelyjä. WPF-tyylimäärittelyt muistuttavat www-sivuilla käytettyä CSS-tyylikieltä, jossa kerrotaan mitä ulkoasumäärittelyksiä eri elementtiryhmillä tai -luokilla tehdään. Tyylimäärittelyllä saadaan koko sovelluksen ulkoasu yhtenäiseksi ilman jokaisen yksittäisen elementin ulkoasun säätämistä erikseen. (Mosers 2011; Ekonoja – Lahtonen 2012.)

WPF:llä toteutetussa käyttöliittymässä käyttöliittymä toteutetaan XAML:llä ja käyttöliittymän toiminnallisuus kehittäjän oman valinnan mukaan joko C#:lla tai Visual Basicilla. Tämä mahdollistaa käyttöliittymän käyttämisen monissa ympäristöissä. Koska näkymä ja toiminnallisuus on erotettu toisistaan, voidaan käyttöliittymän ulkoasua muuttaa helposti. (Kallonen 2008.)

### **6.2.5 XAML**

Extensible Application Markup Language (XAML) on Microsoftin kehittämä yksinkertainen XML-kieleen perustuva kieli, jota käytetään mm. WPF:ssa, Silverlightissa ja Windows Workflow Foundation (WF) -tekniikoissa. XAML:ää käytetään WPF:ssä käyttöliittymän merkintäkielenä määrittelemään käyttöliittymäelementit, datasidokset, tapahtumat ja muut ominaisuudet. (Mosers 2011.)

### **6.3 Effic-aalusta-arkkitehtuuri**

*Seuraavaksi on kuvattu tarkemmin Effic-aalusta-arkkitehtuuri. Koska se on salaista tietoa, se on poistettu tästä julkisesta versiosta.*

### **6.4 Effic-a-komponenttiarkkitehtuuri**

*Seuraavaksi on kuvattu tarkemmin Effic-a-komponenttiarkkitehtuuri. Koska se on salaista tietoa, se on poistettu tästä julkisesta versiosta.*

### **6.5 Anamneesikomponentin arkkitehtuuri**

Anamneesin arkkitehtuuri tulee toteuttaa vastaavalla tavalla kuin toteutetaan valmisteilla oleva Dental Status -komponentti. Kehitysympäristönä on Visual Studio, ohjelmointikielenä C# ja kontrollit toteutetaan WPF:llä. Toteutuksen tulee noudattaa ohjelmointistandardeja, niiltä osin kuin se on mahdollista. Anamneesi-sovelluksen perusrakenteen tulisi olla kuvan 47 kaltainen. (Status Architecture. 2012.)

*Seuraavaksi on kuvattu tarkemmin anamneesikomponentin arkkitehtuuri. Koska se on salaista tietoa, se on poistettu tästä julkisesta versiosta.*

## **7 ANAMNEESIKOMPONENTIN TIETOKANTASUUNNITTELMA**

### **7.1 Tietokannat**

Tietokanta on jotakin käyttötarkoitusta varten luotu tietovarasto. Tietokantaan tallennetaan tietoja, joilla on merkitystä käyttäjälle ja joita käyttäjä haluaa myöhemmin käsitellä. Tietokantoja sisältyy melkein jokaiseen ohjelmistoon, joihinkin jopa kymmeniä. Tietokanta voi olla pieni ja sisältää vain yhden taulun tai se voi olla suuri ja monimutkainen. (Alaluukas 2010.)

Tietokanta voidaan jakaa käyttäjille usealla eri tavalla. Henkilökohtaisessa tietokannassa tietokanta sijaitsee käyttäjien omilla koneilla. Keskuskonepohjaisessa järjestelmässä tietokanta ja sovellukset sijaitsevat keskuskoneella ja sitä käytetään ns. tyhmiltä päätteiltä. Tiedostopalvelinmallissa tietokanta on tiedostopalvelimella ja sovellusohjelmat käyttäjien koneella. Asiakas-palvelinmallissa tietokanta on palvelimella ja sovellusohjelmat työasemilla. Asiakas-palvelinmallia käytetään yleisesti suuremmissa ohjelmistoissa, koska se on suorituskykyisempi, turvallisempi ja luotettavampi. (Lipsanen 2011.)

### **7.2 Relaatietietokanta ja normalisointi**

Erilaisista tietokantamalleista relaatiotietokanta on yleisimmin käytetty tietokantamalli. Relaatietietokannat muodostuvat yhteen liitetyistä tauluista, jotka koostuvat riveistä ja sarakkeista, joilla tiedot esitetään. Kullekin taululle annetaan perusavain ja lapsitauluille myös viiteavain. Perusavaimella taulun rivit yksilöidään, joten arvon tulee olla yksilöllinen eikä se saa puuttua. Viiteavaimella lapsitaulu sidotaan äititauluun ja luodaan yhteys näiden kahden taulun välille. (Alaluukas 2010.)

Hyvä tietokanta tukee tietojen hakemista, sisältää tehokkaasti rakennettuja taulurakenteita, huolehtii tietojen eheydestä sekä sen rakenteen muokkaaminen ja ylläpito on helppoa. Hyvä tietokanta tukee normalisointia. Normalisoitu relaatiotietokannan rakenne tukee tietojen ehjää tallennusta ja tiedonhaun tehokkuutta.

Normalisoinnilla vähennetään tiedon toistumista ja siitä aiheutuvia ongelmia tietojen lisäämisessä, päivityksessä ja poistamisessa. Normalisointi lisää rakenteiden selkeyttä, yhtenäisyyttä ja laajennettavuutta. Normalisointi onkin tietojen järjestämisen prosessi, johon sisältyy taulukoiden luomista ja taulukoiden välisten suhteiden järjestämistä sääntöjä noudattaen. Normalisointiin löytyy useita erilaisia tapoja ja sääntöjä. Nämä tavat ja säännöt on suunniteltu sekä suojaamaan tietoja että tekemään tietokannasta entistä joustavampi poistamalla tiedon toistumista ja epäyhtenäiset riippuvuussuhteet. (Alaluukas 2010; Tietokannan normalisointi. 2012; Tietokannan normalisoinnin perusteiden kuvaus. 2008.)

### **7.3 Microsoft SQL Server ja SQL**

Microsoftin kehittämä SQL Server on relaatiotietokannan hallintajärjestelmä, joka hoitaa asiakkaan ja palvelimen välisen palveluliikenteen SQL-kielellä. Ohjelmiston ensisijainen tehtävä on tietojen tallennus ja hakeminen pyynnöstä muihin sovelluksiin. Tiedot voivat olla samalla koneella tai toisella koneella, jolloin tiedot haetaan verkon kautta. (Microsoft SQL Server. 2011.)

Structured Query Language (SQL) on standardoitu kyselykieli, jolla voidaan tehdä erilaisia hakuja, muutoksia ja lisäyksiä relaatiotietokantaan. SQL sisältää myös sisäänrakennettuja funktioita, joilla voidaan mm. kysellä sarakkeissa olevien tietojen piirteitä sekä ympäristömuuttujia. (SQL. 2011.)

### **7.4 Anamneesin tietokanta**

Ohjelman tietokanta toteutetaan MS SQL Server 2008:lla. Anamneesin tauluista tuli olla liitokset Effican tietokannassa valmiina oleviin taulurakenteisiin. Tietojen tallennukseen tuli suunnitella rakenteet, joihin voitiin tallentaa Ruotsin potilastietolain vaatimat tietojen käytön seurantatiedot. Tietokannan tuli noudattaa normalisointia.

Tietokannan rakennekuviissa ovat ne sarakkeet lihavoituna, joiden arvot ovat not null -arvoja. Kun sarakkeen arvo on not null, silloin sarake ei voi olla tyhjä,

vaan siihen tulee aina tallentaa jokin arvo. Pääavain on merkitty PK (primary key) ja viiteavain FK (foreign key). Osassa rakennekuvista puuttuu viiteavainmerkintä, koska ER-kaavio olisi tullut sekavaksi, jos kaikki relaatiosuhteet olisi siihen merkitty.

*Seuraavaksi on kuvattu tarkemmin anamneesin tietokantaa ja tietokantatauluja. Koska ne ovat salaista tietoa, ne on poistettu tästä julkisesta versiosta DENAnamnesisHealthDataRow-taulun rakennekuvausta lukuun ottamatta.*

### **DENAnamnesisHealthDataRow**

DENAnamnesisHealthDataRow-tauluun (kuva 14) tallentuvat Healthy-, No medicine- ja No allergy -valintaruutujen tiedot. Viiteavaimina ovat AnamnesisRID, joka viittaa DENAnamnesis-tauluun liittäen tiedot potilaan henkilötietoihin ja AnamnesisDateRID, joka viittaa DENAnamnesisDateRow-tauluun liittäen nämä tiedot tiettyyn päiväysriviin. Sarakkeet HealthAnswer, MedicineAnswer ja AllergyAnswer ovat bit-tietotyyppiä. Kun valintalaatikko on valittuna, tallentuu sitä vastaavaan sarakkeeseen tiedoksi 1, ja jos se ei ole valittuna, tallentuu siihen 0.

DENAnamnesisHealthDataRow		
<b>PK</b>	<b>RID</b>	<b>int</b>
<b>FK2</b>	<b>AnamnesisRID</b>	<b>int</b>
<b>FK1</b>	<b>AnamnesisDateRID</b>	<b>int</b>
	<b>HealthAnswer</b>	<b>bit</b>
	<b>MedicineAnswer</b>	<b>bit</b>
	<b>AllergyAnswer</b>	<b>bit</b>
<b>FK3</b>	<b>CreatorRID</b>	<b>int</b>
	<b>CreateDate</b>	<b>datetime</b>
	ModifierRID	int
	ModifyDate	datetime

*KUVA 14. DENAnamnesisHealthDataRow-taulun rakenne*

## 8 LOPUKSI

Anamneesikomponentin suunnittelu onnistui hyvin. Asiakkaat olivat innostuneita suunnitelluista toiminnallisuuksista ja odottavat innolla sen toteutusta. Koska anamneesikomponentti toteutetaan myöhemmin, joihinkin anamneesin arkkitehtuuriratkaisuihin voidaan tehdä muutoksia, jos niihin on silloin tarjolla uudempia ja parempia ratkaisuja.

Työn aluksi minut yllätti se, kuinka monet lait ja direktiivit sekä standardit vaikuttivat terveydenhuollon ohjelmistoihin ja kuinka monenlaisia dokumentteja esimerkiksi MDD vaatii, ennen kuin ohjelmisto voidaan tuoda markkinoille. Tähän työhön ei kuitenkaan kuulunut kaikkien näiden dokumenttien teko, koska osa dokumenteista tehdään vasta toteutuksen jälkeen. Jos tämä komponentti liitetään johonkin vuosiversioon, voidaan tämän sovelluksen dokumentit yhdistää vuosiversion dokumentteihin.

Eniten aikaa vaatinut suunnitteluosa tämän työn toteutuksessa oli NPÖ Blomman käytön suunnittelu. Koska NPÖ Blommaa ei ollut käytetty Efficassa aikaisemmin, oli yksi tärkeimmistä suunnittelutehtävistä selvittää, mitä Nationell Patientöversiktin (NPÖ) suositukset sisältävät ja mitä siihen kuuluva Blommasymboli merkitsee sekä miten se voidaan ottaa käyttöön suunnittelussa anamneesissa. Selvityksen aikana selvisi, että tietojen välittämisen ja hakemisen suunnittelu NPÖ:stä on niin suuritöinen, ettei opinnäytetyöhön varattu aika riitä siihen, joten päätimme aluksi toteuttaa NPÖ Blomman vain anamneesista päivityväksi.

Kaiken kaikkiaan työ on ollut sekä haastavaa että mielenkiintoista. Komponentin arkkitehtuurin suunnittelu ja nimenomaan sen ymmärtäminen oli vaativaa. Koska Efficassa on laaja ohjelmisto ja sen arkkitehtuuri on monimutkainen, sen sisäistäminen vaati aikaa ja paneutumista. Kielellisen kehitykseni kannalta tämä työ on ollut todella hyödyllinen, koska olen saanut tämän työn aikana käyttää suomen lisäksi englantia ja ruotsia, kun aikaisemmissa töissäni olen selvinnyt suomeksi.



## LÄHTEET

.NET Framework. 2011. Wikipedia. Saatavissa:

[http://fi.wikipedia.org/wiki/.NET\\_Framework](http://fi.wikipedia.org/wiki/.NET_Framework). Hakupäivä 13.10.2011.

.NET Framework and its architecture. 2011. Saatavissa: [http://hassan-](http://hassan-baig.blogspot.com/2011/03/net-framework-and-its-architecture.html)

[baig.blogspot.com/2011/03/net-framework-and-its-architecture.html](http://hassan-baig.blogspot.com/2011/03/net-framework-and-its-architecture.html). Hakupäivä 13.10.2011.

Alaluukas, Pekka 2010. T740703 Tiedonhallinta 3 op. Opintojakson oppimateriaali syksyllä 2010. Oulu: Oulun seudun ammattikorkeakoulu, tekniikan yksikkö.

Berg, Christer 2009. The Swedish National Patient Overview (NPO), Background and status. Saatavissa:

<http://www.intersystems.fi/assets/media/web/0/1777.pdf>. Hakupäivä 31.1.2012.

DT 4.1 Dagant Solution. 2011. Tieto Oyj. Ei julkinen.

Dental Architecture Guideline. 2010. Tieto Oyj. Ei julkinen.

Effica Laaturekisteri - Tuotekuvaus. 2011. Tieto Oyj. Ei julkinen.

Ekonoja, Antti – Lahtonen, Tommi 2012. Windows Presentation Foundation (WPF) – Luento 12. Saatavissa: <http://appro.mit.jyu.fi/gko/luennot/luento12/>.

Hakupäivä 18.2.2012.

Fagerberg, Christina 2011. Nationell Patientöversikt (NPÖ). Saatavissa:

<http://www.vgregion.se/upload/Patientdatalagen/Nationell%20Patient%D6versikt%20-%20NP%D6.pdf>. Hakupäivä 12.3.2012.

IEEE. 2011. Saatavissa: <http://fi.wikipedia.org/wiki/IEEE>. Hakupäivä

13.10.2011.

Haikala, Ilkka – Märijärvi, Jukka 2006. Ohjelmistotuotanto. Helsinki: Talentum Media Oy.

Health informatics - Alert information in health records - Concepts, information model and visual presentation. 2009. Saatavissa:

<http://sites.google.com/site/warninginformation/>. Hakupäivä 31.1.2012

Hiltunen, Maarit. Johdatus tietojärjestelmiin, 2 ov. Verkkokurssin oppimateriaali. Oulu. Oulun kauppaoppilaitos. Saatavissa:

[http://www.okol.org/verkkokurssit/datanomi/tietojarjestelmien\\_kaytto\\_ja\\_kehittaminen/johdatus\\_tietojarjestelmiin/kehittamistyyn\\_vaiheet\\_ja\\_elikaarimallit/kehittamistyyn\\_vaiheet\\_ja\\_elikaarimallit.htm](http://www.okol.org/verkkokurssit/datanomi/tietojarjestelmien_kaytto_ja_kehittaminen/johdatus_tietojarjestelmiin/kehittamistyyn_vaiheet_ja_elikaarimallit/kehittamistyyn_vaiheet_ja_elikaarimallit.htm). Hakupäivä 11.2.2012.

Kallonen, Jari 2008. Windows Presentation Foundation –perusteet. Saatavissa:

[http://www.google.fi/url?sa=t&rct=j&q=windows%20presentation%20doundation%20%E2%80%93perusteet&source=web&cd=4&ved=0CEQQFjAD&url=http%3A%2F%2Fdownload.microsoft.com%2Fdownload%2F4%2F7%2Fa%2F47ae4e2b-c576-4ba9-b2d7-5b93ef30206d%2FDevDays2008\\_WPF\\_perusteet.ppt&ctbs=lr%3Alang\\_1fi&ei=97w\\_T\\_65OrTT4QT44f2OCA&usq=AFQjCNG2c1G48eFZQBBmRUa31i\\_8utKLeQ&cad=rja](http://www.google.fi/url?sa=t&rct=j&q=windows%20presentation%20doundation%20%E2%80%93perusteet&source=web&cd=4&ved=0CEQQFjAD&url=http%3A%2F%2Fdownload.microsoft.com%2Fdownload%2F4%2F7%2Fa%2F47ae4e2b-c576-4ba9-b2d7-5b93ef30206d%2FDevDays2008_WPF_perusteet.ppt&ctbs=lr%3Alang_1fi&ei=97w_T_65OrTT4QT44f2OCA&usq=AFQjCNG2c1G48eFZQBBmRUa31i_8utKLeQ&cad=rja). Hakupäivä 18.2.2012.

Kallonen, Jari 2011. WPF ja MVVM -malli. Saatavissa:

<http://video.fi.msn.com/watch/video/wpf-ja-mvvm-jari-kallonen-tieturi/15d3c8amw.%20>. Hakupäivä 13.2.2012.

Knuutila, Jari 2010. Ohjelmistot ja lakiuudistus. Saatavissa:

[http://www.valvira.fi/files/TLTesitykset/Ohjelmisto\\_ja\\_lakiuudistus\\_Knuutila.pdf](http://www.valvira.fi/files/TLTesitykset/Ohjelmisto_ja_lakiuudistus_Knuutila.pdf).

Hakupäivä 15.10.2011.

Koskimies, Kai – Mikkonen, Tommi 2005. Ohjelmistoarkkitehtuurit. Helsinki: Talentum Media Oy.

Kuikka, Seppo 2012. ACI-32040 Automaation ohjelmistokomponentit ja sovelluspalvelut 7 op. Opintojakson luentokalvot. Tampere: Tampereen teknillinen yliopisto, Systemiteknikan laitos. Saatavissa:

[http://www.ac.tut.fi/aci/courses/ACI-32040/Kompo\\_luennot.htm](http://www.ac.tut.fi/aci/courses/ACI-32040/Kompo_luennot.htm). Hakupäivä 12.2.2012.

Käyttöliittymä. 2011. Wikipedia. Saatavissa:

<http://fi.wikipedia.org/wiki/K%C3%A4ytt%C3%B6liittym%C3%A4>. Hakupäivä 18.2.2012.

L 629/2010. Laki terveydenhuollon laitteista ja tarvikkeista. Saatavissa:

<http://www.finlex.fi/fi/laki/alkup/2010/20100629>. Hakupäivä 15.10.2011.

Lahti, Anne 2011. R&D Team Manager, Tieto Nordic Healthcare Dental R&D. Haastattelu 5.10.2011.

Laine, Harri 2011. 581358 Ohjelmistoarkkitehtuurit 5 op. Luentomateriaalit. Helsinki: Helsingin yliopisto, Tietojenkäsittelytieteen laitos. Saatavissa:

<http://www.cs.helsinki.fi/node/65697>. Hakupäivä 12.2.2012.

Lipsanen, Sari 2011. K1042TK Tietokannan hallinta 3 op. Opintojakson oppimateriaali syksyllä 2011. Oulu: Oulun seudun ammattikorkeakoulu, liiketalouden yksikkö.

Luukkainen, Matti 2012. Ohjelmistotuotanto Luento 2. Saatavissa:

<http://www.cs.helsinki.fi/group/java/k12-ohtu/luento2.pdf>. Hakupäivä 20.3.2012.

Lövström, Rikard 2009. Medical Alerts. Saatavissa:

<http://sites.google.com/site/warninginformation/>. Hakupäivä 31.1.2012.

MCV-arkkitehtuuri. 2011. Wikipedia. Saatavissa:

<http://fi.wikipedia.org/wiki/MVC-arkkitehtuuri>. Hakupäivä 11.2.2012.

Microsoft Application Architecture Guide, 2nd Edition. 2009. Saatavissa:

<http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=16236>.

Hakupäivä 4.10.2011.

Microsoft SQL Server. 2011. Wikipedia. Saatavissa:

[http://en.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://en.wikipedia.org/wiki/Microsoft_SQL_Server). Hakupäivä 13.10.2011.

Tietokannan normalisoinnin perusteiden kuvaus. 2008. Microsoftin tuotetuki. Saatavissa: <http://support.microsoft.com/kb/283878/fi>. Hakupäivä 25.4.2012.

Model View ViewModel. 2012. Wikipedia. Saatavissa: [http://en.wikipedia.org/wiki/Model\\_View\\_ViewModel](http://en.wikipedia.org/wiki/Model_View_ViewModel). Hakupäivä 11.2.2012.

Mosers, Christian 2011. WPF Tutorial.net. Saatavissa: <http://www.wpftutorial.net/>. Hakupäivä 18.2.2012.

Mäkelä, Kari 2006. Terveydenhuollon tietotekniikka: Terveyden ja hyvinvoinnin sovellukset. Helsinki: Talentum Media Oy.

Nousiainen, Eero 2009. T740605 Ohjelmistotuotanto 5 op. Opintojakson oppimateriaali syksyllä 2009. Oulu: Oulun seudun ammattikorkeakoulu, tekniikan yksikkö.

NPÖ – Uusi tapa käsitellä potilastietoja. Saatavissa: <http://www.orebroll.se/Files-sv/%C3%96rebro%20%C3%A4ns%20landsting/V%C3%A5rd%20och%20h%C3%A4lsa/NP%C3%96/Finska.pdf?epslanguage=sv>. Hakupäivä 31.1.2012.

Ohjelmistotuotanto. 2012. Wikipedia. Saatavissa: <http://fi.wikipedia.org/wiki/Ohjelmistotuotanto>. Hakupäivä 11.2.2012

Pöyhönen, Ilpo 2010. Terveydenhuollon tuotteen ohjelmistot. Saatavissa: [http://www.google.fi/url?sa=t&source=web&cd=7&ved=0CFwQFjAG&url=http%3A%2F%2Fwww.turkuamk.fi%2Fpublic%2Fdownload.aspx%3FID%3D101550%26GUID%3D%257B702D1329-7638-4C36-89BB-C6B7B82968C1%257D&rct=j&q=mdd%20wiki&ctbs=lr%3Alang\\_1fi&ei=OWyZTpmUB4KB4gTDvvmJBA&usg=AFQjCNEpdb7MjZICLubZE4KCKT\\_y5JgPSw&sig2=4TG69ZPknred5tNRt1OWQ&cad=rja](http://www.google.fi/url?sa=t&source=web&cd=7&ved=0CFwQFjAG&url=http%3A%2F%2Fwww.turkuamk.fi%2Fpublic%2Fdownload.aspx%3FID%3D101550%26GUID%3D%257B702D1329-7638-4C36-89BB-C6B7B82968C1%257D&rct=j&q=mdd%20wiki&ctbs=lr%3Alang_1fi&ei=OWyZTpmUB4KB4gTDvvmJBA&usg=AFQjCNEpdb7MjZICLubZE4KCKT_y5JgPSw&sig2=4TG69ZPknred5tNRt1OWQ&cad=rja). Hakupäivä 15.10.2011.

Rahikainen, Markku 2011. C# Windows-ohjelmointi. Tampere: Stickmansoft.

Rouhiainen, Eeva-Kaisa 1997. Käyttöliittymän visuaalinen suunnittelu. Saatavissa: <http://www.mit.jyu.fi/opiskelu/seminaarit/bak/kayttoliittyma/>. Hakupäivä 7.3.2012.

Ruotsin hammashuollon Effica. 2011. Tieto Oyj. Ei julkinen.

Räisänen, Paavo 2011. C# Perusopas. Saatavissa: <http://www.nettilakka.com/Ohjelmoimaan/Perusopas.pdf>. Hakupäivä 12.3.2012.

Salmivalli, Lauri 2010. Liite 2. Selvitys sosiaali- ja terveydenhuollon ICT-palveluiden Euroopan unionin tasoisesta kehittämisestä sekä alan standardeista. Saatavissa: [http://www.vtv.fi/files/2441/217\\_2011\\_Liite\\_2\\_nettiin.pdf](http://www.vtv.fi/files/2441/217_2011_Liite_2_nettiin.pdf). Hakupäivä 31.1.2012.

Sarja, Jari 2006. Relaatiotietokanta. Saatavissa: <http://www.verkkopedagogi.net/vanhat/fi/sisalto/materiaalit/access2003/luku0375c6.html?C:D=419702&selres=419702>. Hakupäivä 7.10.2011.

Shifflett, Karl 2008. Learning WPF M-V-VM. Saatavissa: <http://karlshifflett.wordpress.com/2008/11/08/learning-wpf-m-v-vm/>. Hakupäivä 14.2.2012.

Sinkkonen, Irmeli 2004. Käyttöliittymät ja käytettävyys. Saatavissa: <http://www.adage.fi/blogi/2004/kayttoliittymat-ja-kaytettavyys/>. Hakupäivä 18.2.2012.

SOSFS 2008:14. Socialstyrelsens föreskrifter om informationshantering och journalföring i hälso- och sjukvården. Saatavissa: <http://www.socialstyrelsen.se/sosfs/2008-14>. Hakupäivä: 7.3.2012.

SQL. 2011. Wikipedia. Saatavissa: <http://en.wikipedia.org/wiki/SQL>. Hakupäivä 13.10.2011.

Status Architecture. 2012. Tieto Oyj. Ei julkinen.

Suun terveydenhuolto - Tuotekuvaus. 2010. Tieto Oyj. Ei julkinen.

Taina, Juha 2000. Ohjelmistotuotannon prosessimallit. Saatavissa: <http://www.cs.helsinki.fi/u/taina/ohtu/k-2001/luennot/prosessi/kaikki.html>. Hakupäivä 11.2.2012.

Tieto 2 minuutissa. 2011. Saatavissa: <http://www.tieto.fi/tiedosta/tieto-2-minuutissa>. Hakupäivä 4.1.2012.

Tietokannan normalisointi. 2012. Wikipedia. Saatavissa: [http://fi.wikipedia.org/wiki/Tietokannan\\_normalisointi](http://fi.wikipedia.org/wiki/Tietokannan_normalisointi). Hakupäivä 9.3.2012.

Windows Presentation Foundation. 2011. Wikipedia. Saatavissa: [http://en.wikipedia.org/wiki/Windows\\_Presentation\\_Foundation](http://en.wikipedia.org/wiki/Windows_Presentation_Foundation). Hakupäivä 18.2.2012.

Vuori, Matti – Kivistö-Rahnasto, Jouni - Toivonen, Sirra 1998. Hyvä käyttöliittymäsuunnittelu lähtee käytön tarpeista. Saatavissa: <http://www.vtt.fi/inf/julkaisut/muut/1998/av7-98.pdf>. Hakupäivä 18.2.2012.