

Opinnäytetyö (AMK)

Tietotekniikan ko.

Internet-tekniikka

2010

Esa Aaltonen

SÄHKÖINEN TUTKIMUSTIETOJEN KERÄÄMINEN JA KÄSITTELY

– eCRF-sovellus potilastutkimukseen



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietotekniikan ka. | Internet-tekniikka

Kesäkuu 2010 | 26 sivua

Ins. Olli Ojala ; TkL Juha Nikkanen

Esa Aaltonen

SÄHKÖINEN TUTKIMUSTIETOJEN KERÄÄMINEN JA KÄSITTELY

Tässä työssä tutustuttiin paperisten ja sähköisten potilastietojenkeruulomakkeiden ominaisuuksiin ja tutkittiin, miten paperinen versio eroaa sähköisestä. Käytiin läpi sähköisen version edut, taloudelliset hyödyt ja haitat verrattuna paperiseen ja verrattiin 4:ä eri sähköistä lomaketta toisiinsa. Lopulta käytiin läpi, miten BASE-ACS-tutkimuksen paperinen versio muutettiin sähköiseksi.

Sähköisen lomakkeen edut paperiseen verrattuna ovat ajan ja kustannusten säästö, lomakkeen käsittelyjen väheneminen, tilan säästö ja kerätyn datan laadun paraneminen.

Sähköisten lomakkeiden vertailussa huomattiin, että kaikki lomakkeet olivat aika samanlaisia. Lomakkeet olivat kaikki aika monipuolisia, mutta parhaiten soveltuva lomake saadaan tekemällä sellainen tutkimusta varten.

BASE-ACS-tutkimuksen paperinen versio koostui n. 25 sivusta kysymyksiä. Nämä kysymykset siirrettiin MySQL-tietokantaan, josta ne pystyttiin PHP-ohjelma lukemaan selaimella käytettäviksi sähköisiksi lomakkeiksi. Kysymykset muodostivat 2 eri lomaketta: pääosion ja followup-osion.

PHP-ohjelma koostui 11 tiedostosta, jotka oli alusta asti tehty BASE-ACS-tutkimusta varten. Varsinaiseen PHP-koodiin ei työssä juurikaan perehdytty.

Lomakkeiden vastaukset siirrettiin kantaan, josta ne pystyttiin lukemaan selaimella PHP-ohjelmalla käsiteltynä CSV-tiedostoon.

ASIASANAT:

potilasasiakirjat, sähköiset lomakkeet, tietokannat, tietojenkäsittely

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information Technology | Internet Technology

June 2010 | 26 pages

Olli Ojala, B.Eng. ; Juha Nikkanen Lic.Tech., Principal Lecturer

Esa Aaltonen

ELECTRONIC CASE REPORT FORM

This thesis explains the features of regular and electronic case report forms (eCRFs) and tells how they differ from each other, taking a look at eCRF's strengths and weaknesses compared to a paper CRF. Finally the thesis shows how the paper version of the CRF in BASE-ACS study was converted into the electronic version.

The strengths of eCRF are saving time and costs and reducing the need deal with queries. Also saving of physical space and improving the quality of the data are eCRF's features.

The comparing of a few different eCRFs revealed that there was not much of a difference when it comes to features.

The paper version of the BASE-ACS study had about 25 pages of questions. These questions were typed into a MySQL database, from where it was easy to read them to an electronic form. A browser based PHP program was created to make the form. The questions formed two parts: main part and follow-up part.

The PHP program consist of 11 files. All of them had been made for the BASE-ACS eCRF, which remained fairly simple as far as eCRF's go.

Answers to the form questions were saved to a database, from where they could be exported using the PHP program to a CSV file.

KEYWORDS: patient reports, electronic forms, databases, data processing

SISÄLTÖ

SISÄLTÖ	IV
SYMBOLIT JA LYHENTEET	V
1 JOHDANTO	1
2 POTILASTIETOJENKERUULOMAKE	2
2.1 Yleistä potilastietojenkeruulomakkeista	2
2.2 Lomakkeen suunnittelu	3
3 SÄHKÖINEN CRF	3
3.1 Paperilta sähköiseksi	4
3.2 Testatut valmiit ohjelmistot	6
4 TUTKIMUSTIETOLOMAKE (BASE-ACS)	9
4.1 Suunnittelu ja versiot	9
4.2 Peruslomake	11
4.3 Followup-sivut	12
4.4 Muut osat	12
4.5 Tietokanta	14
5 SÄHKÖISEN LOMAKKEEN OHJELMALLINEN TOTEUTUS	15
6 TESTAUS JA KÄYTTÖÖNOTTO	18
7 YHTEENVETO	20
LÄHTEET	21
KUVAT	
Kuva 1. eCRF:n ja paperilomakkeen eroja	6
Kuva 2. Ohjelman päävalikko	13
Kuva 3. Ohjelman kulkukaavio	14
Kuva 4. Sivun mukana liikkuva valikko	18
TAULUKOT	
Taulukko 1. eCRF:n ominaisuudet ja vahvuudet	4
Taulukko 2. eCRF-ohjelmien vertailun tulokset	8

SYMBOLIT JA LYHENTEET

CRF	potilastietojenkeruulomake (Case Report Form)
eCRF	sähköinen potilastietojenkeruulomake (electronic Case Report Form)
EDC	sähköinen tiedon tallennus (Electronic Data Capture)
HTML	merkintäkieli, josta internet-sivut koostuvat (Hypertext Markup Language)
MySQL	relaatiotietokantojen hallintajärjestelmä
PHP	ohjelmointikieli, jolla saadaan HTML-sivuista tehtyä dynaamisia (Hypertext Preprocessor)

1 JOHDANTO

Lääketieteellisen tutkimuksen tavoitteina on muun muassa kehittää diagnostiikkaa, parantaa potilashoitoa ja hoidon tuloksia. Lääketieteellisessä tutkimuksessa kerätään potilastietoja selvittämään esim. sairauksien ilmenemismuotoja, hoidon vaikuttavuutta ja kustannustehokkuutta. Tutkimustietoja kerätään tutkimusta varten suunnitelluilla potilastietokeruulomakkeilla. Kerätyn tiedon määrä on usein suuri ja kerääminen on helpompaa sähköisillä välineillä kuin esim. perinteisillä paperisilla versioilla. Toisinaan pitää olemassa oleva paperinen versio muuttua sähköiseksi. Paperisille lomakkeillekin kerätty raaka tieto pitää myöhemmin siirtää esim. Excel-taulukon ja tilastointiohjelman analysointia varten.

Opinnäytetyössä perehdytään potilastietojenkeruulomakkeisiin ja tarkastellaan, mitä etuja ja haittoja sähköiset lomakkeet tuovat. Tarkastellaan myös valmiita sähköisiä lomakkeita ja verrataan niitä työtä varten tehtyyn omaan sähköiseen lomakkeeseen. Vastaavanlaisia vertailuja ei ole tehty, ainakaan julkisesti. Käydään myös läpi oman lomakkeen eri vaiheet suunnittelusta testaukseen. Työssä ei perehdytä PHP-kielen ja MySQL-tietokantojen toimintaan.

Työnä oli rakentaa sähköinen potilastietojenkeruulomake olemassa olevan paperisen pohjalta. Sen tuli helpottaa ja nopeuttaa lomakkeen parissa toimivien työskentelyä. Työn tilaaja oli Satakunnan sairaanhoitopiiri ja lomakkeen aihe "BASE-ACS trial", jonka alaotsikko on vapaasti suomennettuna *bioaktiivisen stentin vertaaminen Everolimusta uuttavaan stenttiin akuutissa sepelvaltimoiireyhtymässä*. Tutkimuksessa vertaillaan kyseisen taudin saaneiden potilaiden tietoja.

2 POTILASTIETOJENKERUULOMAKE

2.1 Yleistä potilastietojenkeruulomakkeista

Potilastietojenkeruulomake eli CRF on pääsääntöinen tapa kliinisissä tutkimuksissa kerätä potilailta tietoja, joita käytetään selvittämään esim. sairauksien vaikutuksia ja yhtäläisyyksiä suuressa joukossa potilaita lokaalisti tai eri puolilla maapalloa. Tutkimuksiin tarvitaan yleensä tietoja isolta joukolta potilaita, koska näin varmistutaan tulosten eheydestä, mutta pienemmälläkin määrällä saadaan suuntaa antavia tuloksia. [1]

CRF-lomakkeille kerättävä tieto riippuu tutkimusasetelmasta ja asetetusta tutkimusongelmasta. Potilailta kerätään yleensä perustiedot, kuten ikä, sukupuoli ja rotu, ja lisäksi tutkimusongelmaan liittyviä valikoituja tietoja potilaan sairaudesta ja hoidosta. Tutkimus voidaan toteuttaa joko yhdessä keskuksessa tai voidaan tehdä yhteistyötä useamman yksikön/sairaalan kanssa ja tehdä ns. monikeskustutkimus. Monikeskustutkimuksissa tieto kerätään sovitusti yhteen, ns. koordinoivaan keskukseseen. Tiedonkeruun jälkeen data syötetään yleensä tietokoneelle tai jonkinlaiseen sähköiseen järjestelmään, jossa niitä voidaan vertailla ja analysoida. [2]

Tietojen käsittely voi olla työläs prosessi. Ensin tarvitaan potilaita, jotka antavat suostumuksensa tutkimukseen osallistumisesta ja joilta tiedot voidaan suostumuksen perusteella kerätä. Tutkimusten seuranta-aika vaihtelee. Toisinaan potilaat voivat joutua käymään useamman kerran vastaanotolla tai vointia seurataan seurantasoitoin, riippuen tutkimuksesta. Joissakin tapauksissa tietoja kerätään vain sairaalahoitovaiheen ajalta. Tutkimukseen otettava potilasmäärä lasketaan tilastotieteellisin keinoin. Kun tutkimusmateriaali on kerätty, lähdetään analysoimaan aineistoa. Tutkimusmateriaalista voidaan tehdä myös välialalyysejä ja alaryhmäanalyysejä. Tutkinta ja vertailu tapahtuu useasti jollakin tilasto- tai analysointiohjelmistolla. Tutkittaessa tietoja törmätään aina jonkin näköisiin virheisiin, ja mitä enemmän potilaita ja tietoa on sitä

enemmän virheitäkin mukana. Useasti virheet voivat olla yksinkertaisia, mutta niiden selvittämisessä saattaa silti kulua aikaa ja resursseja. Varsinkin paperisissa lomakkeissa saattaa virheitä tulla, koska valmiita vastauksia ja parametreja ei välttämättä ole annettu. Sähköisissä järjestelmissä taas on mm. virhelyöntien vaara. [2-3]

2.2 Lomakkeen suunnittelu

Lomakkeiden suunnittelussa tulee ottaa huomioon useita seikkoja. Kysymysten tulee olla sellaisia, että niillä saadaan oikeaa ja tarkkaa tietoa halutussa muodossa. Yleensä on parempi, mitä vähemmän lomakkeessa on sellaisia kysymyksiä, joihin voi vastata usealla eri tavalla. Monivalintakysymykset ovat hyviä, koska vastausvaihtoehdot on valmiiksi annettu, mutta vastauksia voi silti olla monia. Lomakkeen ja kysymysten tulee olla muotoiltu niin, että lomaketta on helppo seurata ja täyttää. Lomake tulisi myös muotoilla niin, että samoja vastauksia ei tulisi useammasta kysymyksestä, jotta vältetään tulosten vääristymiseltä. [1-2]

Lomakkeita ei yleensä tarvitse luoda tyhjästä, sillä lomakepohjia on olemassa. Yleensä kannattaakin muokata uusi lomake mahdollisesta olemassa olevasta ja hyväksi todetusta pohjasta. Tällä säästetään aikaa, koska ei tarvitse aloittaa tyhjästä, ja kun käytetään samasta pohjasta tehtyjä lomakkeita, se vähentää uuden lomakkeen oppimiskynnystä. [1]

3 SÄHKÖINEN CRF

Sähköisistä potilastietojenkeruulomakkeista käytetään lyhennystä eCRF. Ne ovat periaatteessa samanlaisia kuin CRF-lomakkeet, mutta niiden täyttämiseen ja käsittelemiseen käytetään tietokonetta. Yleensä eCRF:t otetaan käyttöön pitkille lomakkeille, joiden käsitteleminen paperilla olisi hyvin työlästä. Esim. 50-sivuinen lomake paperilla veisi yhtä potilasta kohden sen 50 sivua ja siinä vaiheessa, kun vaikka 100 potilaan tiedot pitäisi jonnekin lähettää, pitäisi mahdollisesti skannata tai faksata 5 000 sivua papereita. Lomakkeiden

varastointi olisi myös epäkäytännöllistä. Paperilomakkeillekin syötetty tieto syötetään jossakin vaiheessa tietokoneelle, viimeistään tulosten tarkastelua varten. Toisinaan kesken tutkimuksen siirrytään papereista elektroniseen muotoon.

Elektronisten lomakkeiden vahvuuksiin kuuluu myös se, että niitä on yleensä nopeampi täyttää ja muokata, lisäksi ne pysyvät siistimpinä. Niille pystytään myös asettamaan tiukat rajat ja vastaukset, joita järjestelmään saa syöttää. Voidaan myös luoda dynaamisempia lomakkeita, jotka voivat muuttua syötetyn tiedon mukaan. Taulukkoon 1 on tiivistetty eCRF:n vahvuudet. [1]

Elektronisia lomakkeita ei ole niin helppo tehdä tai teettää kuin paperisia. Jos halutaan siirtyä elektroniseen muotoon, pitää yleensä tilata maksullinen ohjelmisto sitä varten. Joitakin ilmaisohjelmistoja on myös olemassa. [1]

Taulukko 1. eCRF:n ominaisuudet ja vahvuudet. [4]

eCRF:n ominaisuudet	Vahvuudet paperiseen lomakkeeseen verrattuna
Tuplasyöttöjen torjunta	Säästää aikaa ja kustannuksia
Virheiden tarkistus	Käsittelyjen väheneminen
Sähköinen valvonta ja käsittely	Tulostusten, nitomisien ja postituskulujen poisto, tilan viemisen väheneminen
	Ajan säästö, matkakulujen väheneminen

3.1 Paperilta sähköiseksi

Kun paperista lomaketta vertaa sähköiseen, näkyy sähköisen version edut hyvin. Tietokoneiden vahvuuksiin kuuluvat laskennat ja tiedon järjestely. Kaikki tieto, joka syötetään koneelle, saa nämä vahvuudet käyttöönsä. Sähköiset lomakkeet voivat sisältää virheiden tarkistuksen, joka tarkastelee ristiriitaisuuksia datassa ja pyytää tietoa, jos jotakin tärkeää puuttuu. Virheellinen, puuttuva tai tulkitsematon tieto voi aiheuttaa paperisien lomakkeiden kanssa paljon ongelmia. Sähköisillä lomakkeilla nämä virheet

torjutaan jo syöttövaiheessa ja mm. tutkijoille jää enemmän aikaa itse tulosten tarkisteluun. [5]

Ei ole kuitenkaan realistista, että kaikkia virheitä voitaisiin poistaa. Virheitä on kahta eri lajia: systemaattisia ja satunnaisia. Tieto pitää jossain vaiheessa syöttää koneelle ja syötön aikana voi sattua mm. virhelyöntejä eli satunnaisia virheitä. Paperille kirjoittamisen osaavat kaikki lomakkeita täyttävät, mutta tietotekniikan tuntemuksessa saattaa olla eroja. Lomakkeet saattavat myös vaihdella ulkonäöllisesti ja toiminnallisesti, jolloin jokainen lomake vaatii opettelua. [5]

Sähköisten versioiden oletetaan olevan kätevämpiä kuin paperisten, koska tietokantoihin mahtuu tuhansien potilaiden tiedot, jotka paperilla veisivät hyvin paljon tilaa. Paperi on kuitenkin vielä toistaiseksi liikuteltavampi kuin tietokone. Kannettavan tietokoneen saa otettua mukaan, mutta se vaatii useasti enemmän työtä kuin paperilomakkeen ja kynän ottaminen. [5]

Sähköisten lomakkeiden vastaanotto

Sähköiset lomakkeet ovat olleet saatavilla jo monta vuotta. Voisi jopa ajatella, että siitä saakka kun tietokoneet alkoivat yleistymään. Lääketieteen siirtyminen niitä käyttämään on kuitenkin ollut hidasta useammastakin syystä. Sähköisen tietojen hallinnoinnin ja keräämisen etuja ei ole tarkasteltu tarpeeksi ja tietokoneiden käyttäminen paperisten lomakkeiden apuna ei ole edes välttämättä vaikuttanut hyödylliseltä. Tietoturva on ollut myös yksi huoli sähköisissä lomakkeissa, koska aina ei ole ollut mahdollista taata riittävää tietoturvaa salassa pidettäville asiakirjoille ja tiedonkeruulomakkeille. Tietokonetta ajateltaessa pelkkänä apuvälineenä, sen kaikki hyödyt ei välttämättä käy selville. Muutos on myös välillä pelottava ajatus: jos jokin on toiminut monta vuotta hyvin, miksi sitä pitäisi muuttaa. Muutos myös maksaa ja ilman vakuuksia hyödyllisyydestä moni ei viitsi ottaa riskiä ja sijoittaa. [6-7]

Sähköisen lomakkeen taloudellinen hyöty

Teoria siitä, että sähköinen lomake olisi taloudellisesti kannattavampi kuin paperinen, pohjautuu suurimmalta osalta työtuntien vähentämiseen. Sähköisten järjestelmien käytön tulisi helpottaa lomakkeiden kanssa työskentelyä ja siten vähentää myös niihin käytettyjä työtunteja. Joitakin tutkimuksia taloudellisen hyödyn määrittämiseksi on tehty. Esimerkiksi voidaan ottaa Banikin tekemä tutkimus, jossa verrattiin sähköistä järjestelmää paperiseen viidessä eri maassa, 19:llä eri tutkimuspaikalla. Tuloksista (kuva 1) näkee selvästi, että sähköinen lomake vähentää jonkin verran tutkimuksen kestoa, mutta huomattavasti erilaisten tiedustelujen määrää. [4]



Kuva 1. eCRF:n ja paperilomakkeen eroja. [4]

3.2 Testatut valmiit ohjelmistot

Useimmissa sairaaloissa ja lääketieteellisissä laitoksissa tarvitaan CRF-lomakkeita luultavasti jossain vaiheessa ja yhä useammat käyttävät sähköisiä lomakkeita. Tavallisella internethaulla ei kuitenkaan löytynyt juurikaan valmiita ohjelmistoja testattavaksi. Suurimmat yhtiöt, jotka näitä kauppaavat, myyvät niitä suoraan laitoksille. Electronic Data Capture (EDC) -nimitystä käytetään yleisemmin eCRF:n kaltaisista ohjelmista. Käyttäen tätä hakusanana löytyy joitakin tuloksia. [8]

Openclinica on ilmainen, mutta täysin kelvollinen järjestelmä sähköisten lomakkeiden ylläpitämiseen ja täyttämiseen. Ominaisuuksista ei ole puutetta, vaan niitä on oikeastaan liikaa, koska monen eri osapuolen tarpeet on otettu huomioon. Nappuloita ja ponnausikkunoita on paljon. Tämä tekee Openclinicasta hieman sekavan näköisen EDC-sovelluksen ja järjestelmän opettelu vie oman aikansa. Openclinica on täysin selainpohjainen, eli sitä ei tarvitse asentaa erikseen koneelle, vaan pelkkä selain riittää ja siihen pääsee käsiksi mistä vain. Vahvuudet kiteytettynä ovat ilmaisuus, monikäyttöisyys ja selainpohjaisuus.

Entrypoint Plus on todella kattava ohjelma, jossa voi rakentaa omia yksityiskohtaisia ja ominaisuuksiltaan ja valinnoiltaan rikkaita lomakkeita. Ohjelma sisältää editorin, jolla lomakkeet luodaan. Lomakkeen luonti on selvästi suunniteltu edistyneimmille käyttäjille, koska sen ulkonäkökin muistuttaa jo oikeaa ohjelmointiohjelmaa. Sisältää valmiita pohjia lomakkeiden luomisen helpottamiseen. Käyttää SQL-tietokantoja tiedon tallentamiseen ja kantoihin pääsee käsiksi mistä vain, kunhan asiakasohjelma on asennettu. Järjestelmässä on myös mukana jonkin näköinen lääketietokanta. Hieman vanhahkon näköinen, eli näyttää siltä ettei ole uudistettu vähään aikaan. Vahvuudet ovat laajat ominaisuudet ja lomakepohjat.

Infermed Macro v4 on ominaisuuksiltaan aika pitkälti samanlainen kuin Entrypoint Plus. Ulkoasu on modernimman näköinen, mutta välillä ruutuun mahdutetaan ehkä liikaakin tavaraa kerralla, ja lomakkeesta tulee hieman epäselvä. Tulosten tarkastelua varten kattava selain. Vahvuuksiin kuuluu laajat ominaisuudet ja nykyaikainen käyttöliittymä.

Kaikissa edellä mainituissa ohjelmissa on omat vahvuutensa ja riippuukin käyttäjän tarpeista, että mikä on paras. Vertailulle pitääkin asettaa tietty näkökulma, josta se suoritetaan. Tässä tapauksessa voidaan vertailu suorittaa tässä opinnäytetyössä käsitellyssä olevan BASE-ACS-lomakkeen ehdoilla, eli sen näkökulmasta mietitään vertailun eri kohtia. BASE-ACS:n sähköinen lomake, johon tämä opinnäytetyö pohjautuu otetaan myös mukaan vertailuun.

Vertailun pääkohtina ovat soveltuvuus, muokattavuus, luotettavuus ja hinta. Taulukkoon 2 on koottu yhteenveto tuloksista.

Soveltuvuudeltaan sähköinen BASE-ACS-lomake sopii parhaiten, koska se on tehty alusta asti vastaamaan paperista lomaketta työnantajan tarpeiden mukaan. Kaikki muut järjestelmät vaikuttivat soveltuvan moniin eri kohteisiin, joten ne olisivat myös saattaneet toimia tämän tutkimuksen kanssa.

Muokattavuudeltaan Entrypoint Plus vaikutti parhaimmalta editorinsa kanssa. Kaikki muut hyvinä kakkosina. BASE-ACS-lomake saattaa olla hankalimmin muokattavissa ja vaatii perehtymistä järjestelmään.

Luotettavuuteen kuuluu mm. tietojen säilyminen ja lomakkeen toimiminen. Mikään näistä lomakkeista ei ole absoluuttisesti paras, sillä jokaisessa tiedot varastoidaan palvelimelle, jonka luotettavuus taas riippuu sen hallinnoijasta. Joissakin lomakkeissa on varmuuskopiointitoiminto valmiina, mutta esim. BASE-ACS-lomakkeessa se on hoidettu käyttöjärjestelmän kautta. Openclinican luotettavuus on ehkä huonoin, koska se ei ole käyttäjän hallinnassa.

Hinnaltaan paras on tietenkin ilmainen Openclinica. Toisaalta maksullisesta palvelusta saa välillä esim. teknistä tukea paremmin.

BASE-ACS-tutkimusta varten mikä tahansa näistä ohjelmista olisi saattanut käydä, mutta omaa järjestelmää alettiin tekemään, että saatiin mahdollisimman hyvin soveltuva sähköinen lomake.

Taulukko 2. eCRF-ohjelmien vertailun tulokset. 1 on paras ja 3 huonoin.

	BASE-ACS	Openclinica	Entrypoint Plus	Infermed Macro v4
Soveltuvuus	1	2	2	2
Muokattavuus	3	2	1	2
Luotettavuus	1	2	1	1
Hinta	2	1	2	2

4 TUTKIMUSTIETOLOMAKE (BASE-ACS)

Projektin päämäärä oli muuttaa paperinen potilastietojenkeruulomake sähköiseksi. Alkuperäiseen lomakkeeseen kuuluu noin. 25 sivua kysymyksiä ja joitakin opastussivuja. Lomakkeiden normaali käytötapa sisältää täyttämisen, mahdollisesti kuljetuksen tai lähetyksen jopa maasta toiseen, tietojen muuttamisen sähköiseen muotoon Excelin taulukoiden avulla ja lopulta tietojen tarkastelun jollakin analysointi-ohjelmalla, kuten SPSS:llä. Sähköisen version tulisi poistaa ainakin turhat kuljetukset ja tehdä tietojen syöttämisestä sähköiseen muotoon mielekkäämpää. Sähköinen versio vaati myös hieman dynamiikkaa rakenteeseen, koska jotkin kysymykset saattoivat ilmestyä riippuen toisten kysymysten vastauksista. Esim. leesio- ja stenttikysymykset ja niiden määrä riippui siitä, miten lomakkeessa aikaisemmin oli täytetty.

Järjestelmän ohjelmoimiseen on käytetty PHP-kieltä ja sivujen muotoiluun HTML-kieltä ja CSS-tyylimäärittelyjä.

Projektia varten tehty eCRF-lomake koostuu useammasta osasta. Siihen kuuluu peruslomake-osa, joka sisältää suurimman osan kysymyksistä. Nämä kysymykset ovat lähinnä perustietoa potilaan tilasta ja hänelle tehdyistä leikkauksista ja operaatioista ja jotakin tietoa niiden jälkeisestä tilasta. Toinen osa lomaketta on ns. followup-sivut, joissa käsitellään potilaan tilaa kuukausittain. Järjestelmään kuuluu myös potilastietojen päivityssivu ja pieni ylläpitosivu.

4.1 Suunnittelu ja versiot

Projektin suuntaa mietittiin yleensä yhteisissä palavereissa, joissa käytiin läpi ohjelmaan tehdyt muutokset ja sen edistyminen, ja suunniteltiin miten jatketaan. Ohjelman etenemisessä ei käytetty versionumeroita, mutta se voidaan jakaa eri vaiheisiin ajallisesti palaverien ollessa erottajina. Esim. ensimmäinen versio olisi nimeltään vaikka prototyyppi 1 ja sitä käsittelevän palaverin jälkeen alettiin työstämään prototyyppi 2:ta.

Prototyyppi 1

Ensimmäinen versio, jonka työstäminen aloitettiin oikeastaan tyhjästä. Lomakkeen luominen alkoi miettimällä, missä muodossa vanha lomake kannattaisi siirtää elektroniseksi ja syötetäänkö kaikki lomakkeen kentät tietokantaan, josta ne luettaisiin joka kerta kun lomake luodaan, vai kirjoitetaanko kentät suoraan ohjelmaan. Kannassa niitä on helpompi ylläpitää ja muuttaa ja ohjelman koodi pysyy selkeämpänä, joten tämä tuntui paremmalta vaihtoehdolta. Luotiin tietokantaan taulu, jonne syötetään mm. kentän nimi eli kysymys, tyyppi ja vastaukset jos monivalintakysymys. Aluksi syötettiin muutama kenttä kantaan ja luotiin yksinkertainen ohjelma, joka listaa kaikki tulokset kannasta. Ohjelma hakee nimen, tunnistaa tyyppin ja mahdolliset lisätiedot ja tulostaa sen perusteella yhden kysymyksen lomakkeeseen ja toistaa saman kaikille kannassa oleville riveille. Kantaan piti syöttää lisää tietoa, jotta saatiin lomake näyttämään kunnolliselta. Tämä versio oli oikeastaan demonstraatio siitä, millaiselta lomake tulee näyttämään.

Prototyyppi 2

Muotoilujen kanssa piti hieman pelata, että kentät tulostuivat järkevän näköisesti. Eniten ongelmia aiheutti oikeiden kenttien saaminen oikeiden otsikoiden ja aliotsikoiden alle. Ohjelma muutettiin hakemaan kentät otsikoiden perusteella; otsikoilla ja kentillä on tietty numero kannassa ja kun otsikko haetaan, tulevat mukana myös sen numeroon kuuluvat kentät. Kysymykset jaetaan myös sen mukaan onko kyseessä uusi kysymys vai jatkokysymys aikaisemmalle kysymykselle. Uusi kysymys merkitään kantaan taso yhdeksi ja jatkokysymys taso kahdeksi. Lomakkeen muotoilla ensimmäisen tason kysymykset sijaitsevat vasemmalla ja toisen tason kysymykset oikealla. Tässä versiossa mietitty enemmän lomakkeen ulkonäköä palaverissä saadun palautteen perusteella.

Prototyyppi 3

Koska lomakkeen ulkonäkö alkoi olla kunnossa, alettiin ohjelmoida lisää toimintoja. Kaikki peruskysymykset oli syötetty kantaan, ja seuraavaksi piti

syöttää followup-kysymykset. Followup-lomakkeen työstö hoitui helposti sillä perusrakenne oli jo valmiina. Eniten työtä oli followup-sivun luonnissa. Tällä sivulla näkyy mm. kaikkien potilaiden followup-tila followup-kuukausina.

Prototyyppi 4

Leesio- ja stentti-kysymyksille piti saada dynaamisuutta, jotta ne toimivat halutulla tavalla. Tätä varten piti peruslomakkeen taustalle ohjelmoida toiminnallisuutta. Kirjautumissivu tuli uutena ominaisuutena.

Prototyyppi 5

Tähän versioon tuli enimmäkseen pientä toimintojen hiontaa testausta varten. Aikaisempi testaus oli kaatunut ohjelman toimimattomuuteen Internet Explorer 6 -ympäristössä, mutta se ohitettiin asentamalla Firefox-selaimet testikoneisiin.

Prototyyppi 6

Tällä hetkellä uusin versio. Followup-sivut korjattu ja peruslomakkeen käytettävyyttä parannettu. Peruslomakkeen menu myös muutettu seuraamaan lomaketta, kun sitä vieritetään alas tai ylös.

4.2 Peruslomake

Peruslomake koostuu mm. peruskysymyksistä esim. potilaan tiedoista ja sairauden tilasta, mutta sisältää kaikki alkuperäisen lomakkeen kaikki kysymykset paitsi followup-kysymykset.

Lomakkeelle piti ohjelmoida eräänlaista dynaamisuutta, koska eräiden kysymysten määrä riippui toisten kysymysten vastauksista. Tarkennettuna leesioskysymysten ja stenttikysymysten määrä riippui lomakkeeseen syötetystä leesioiden ja stenttien määrästä. Kysymys, jossa kysyttiin stenttien määrää sijaitsi vielä leesioskysymysten seassa. Tämä vaikutti hieman tietokannan muodostumiseenkin, koska ajateltiin, että olisi parempi luoda omat taulut leesioille ja stenteille.

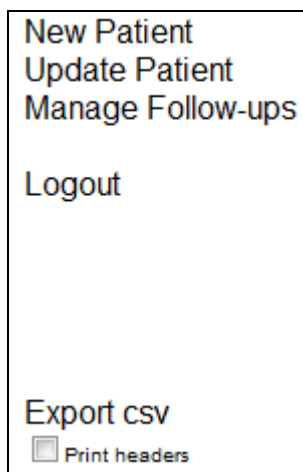
Sivun ylälaitaan ohjelmoitiin eräänlainen laskuri ja pikavalikko, jonka avulla näkee, miten paljon lomakkeesta on täytetty, ja pääsee hyppäämään haluamansa pääotsikon kohdalle. Laskuri käy läpi kannasta kysymysten määrän ja kuinka moneen kysymykseen on vastattu jotakin. Näiden tietojen perustella saadaan jokaisen otsikon alle suuntaa antava numero siitä miten paljon lomaketta on täytetty.

4.3 Followup-sivut

Followup-sivut eli seurantasivut, koostuvat yksinkertaisuudessaan listauksesta, jossa on jokaiselle potilaalle rivi. Riviltä pääsee hyppäämään suoraan potilastietoihin, jos vaikka tarvitsee tarkistaa jotakin. Rivillä näkyy myös potilaan followup-tila 30 päivän, kuuden kuukauden, 12 kuukauden ja 18 kuukauden jälkeen operaation. Ne merkitään joko 'done' tai 'not done', eli tehty tai ei tehty. Nämä ovat linkkejä, josta hypätään followup-lomakkeeseen. Followup-lomake koostuu 36 kysymyksestä ja ne ovat samat jokaiselle followup-ajanhetkelle. Lomakkeen lopussa on valinta, josta valintaan onko sille kuukaudelle followup tehty.

4.4 Muut osat

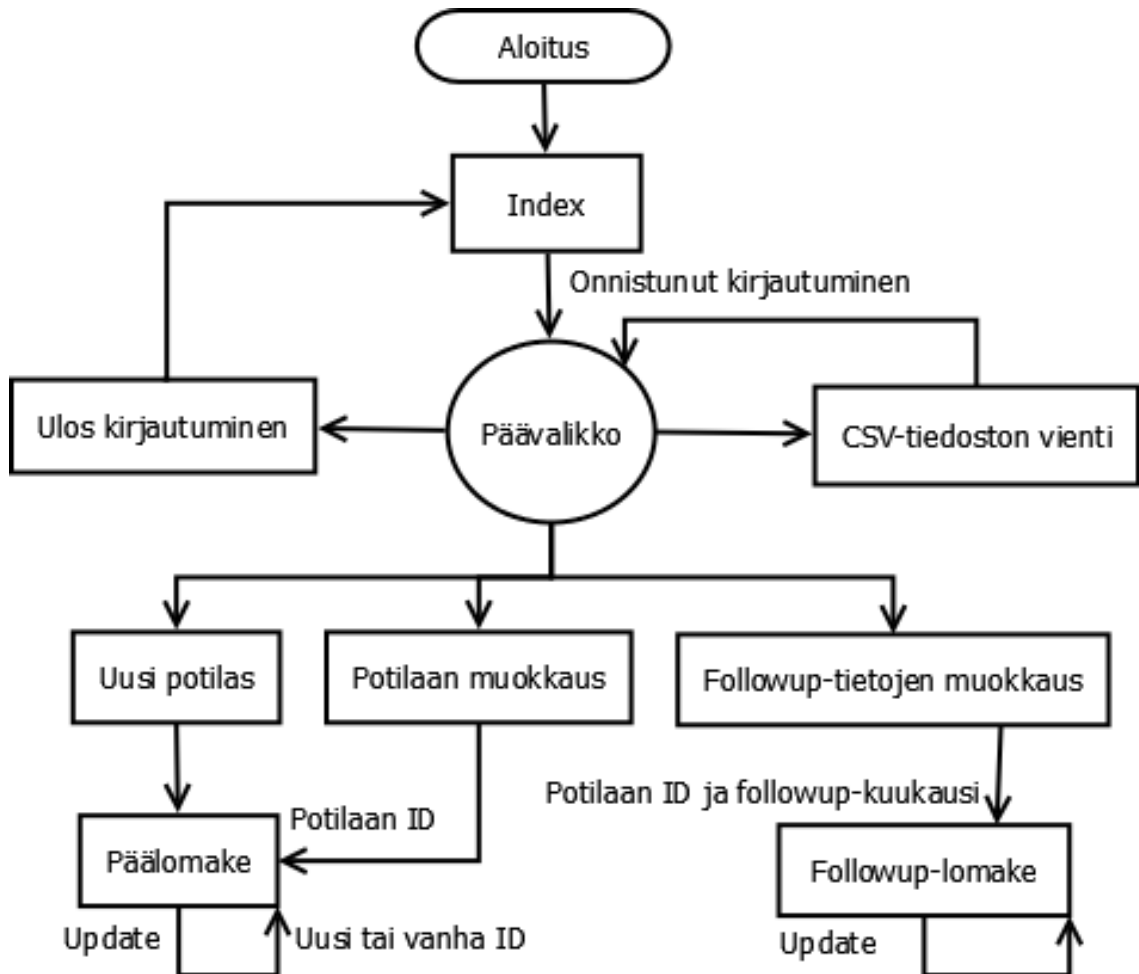
Peruslomakkeen ja followup-sivujen lisäksi lomakkeeseen kuuluu pieni listaussivu, jolla voi tarkastella syötettyjä potilaita ja niiden perustietoja. Tältä sivulta voi myös siirtyä muokkaamaan potilaiden tietoja. Etusivulla kirjautumisen jälkeen tulee näkyviin päävalikko, josta valitaan mitä lomakkeessa halutaan tehdä. (Kuva 2.) Päävalikossa on myös linkki, josta saa ladattua koko tietokannan potilastiedot CSV-muodossa.



Kuva 2. Ohjelman päävalikko.

CSV-tiedostoon kuuluu oikeastaan kaikki potilastiedot kannasta. Lomakkeen tulostus tapahtuu painamalla 'Export CSV' -linkkiä etusivulta. Samassa saa valita, että tulostetaanko CSV-tiedostoon myös otsikot. Linkkiä painamalla, ohjelma hakee potilastiedot 3:sta eri kannassa ja tulostaa ne yhteen tiedostoon.

Kirjautumissivu tulee näkyviin, kun avaa sivun ensimmäistä kertaa, jos PHP-sessio on vanhentunut tai jos on kirjautunut ulos. Kirjautuminen tapahtuu kirjoittamalla kirjautumissivulle käyttäjänimi ja salasana. Ohjelma sen jälkeen vertaa käyttäjänimeä ja salasanan MD5-hashia kannassa oleviin vastaaviin tietoihin. Jos tiedot vastaavat jotakin kannassa olevaa käyttäjänimeä, ohjelma luo uuden PHP-session käyttäjälle. Mahdollisessa tapauksessa, jossa tiedot eivät täsmää, suorittaa ohjelma saman funktion kuin kirjautuessa ulos. Se tyhjentää session tiedoista. Joka kerta kun sivun lataa, tarkistaa ohjelma session tiedoissa olevat käyttäjätiedot ja vertaa niitä kantaan. Sivujen hierarkia laskee alaspäin kirjautumissivulta aina lomakkeen täyttämiseen tai päivittämiseen saakka, kuten kuvassa 3 on näytetty.



Kuva 3. Ohjelman kulkukaavio.

4.5 Tietokanta

Tietokanta muodostuu 9:stä eri taulusta. 4 taulua sisältävät lomakkeen sisältöön liittyvät asiat, kuten peruskysymykset, followup-kysymykset, leesiokysymykset ja otsikot. 4 tauluun tallentuvat lomakkeiden vastaukset. Peruskysymysten, followup-kysymysten, leesiokysymysten ja stenttikysymysten vastaukset. Viimeiseen tauluun on syötetty käyttäjätiedot, salasanat ja oikeudet.

Tietokantana toimii MySQL-tietokanta. Tietokantaan piti syöttää kaikki kysymykset, että saatiin parempi kuva siitä miten lomake tulostuu.

Peruslomakkeeseen kuuluu 165 kysymystä, joista jokaisen kohdalla kantaan piti syöttää vähintään 5 eri tietoa:

- ryhmän numero, jonka avulla kysymys tulostuu oikean otsikon alle
- taso, joka kertoo onko kysymys ns. pääkysymys vai jatkokysymys
- kysymys eli miten se näytetään lomakkeessa
- lomakenimi eli kysymys sellaisessa muodossa, että sitä voidaan käyttää koodin kanssa
- tyyppi eli onko vastaus tekstiä vai monivalinta
- monivalintakysymyksiin vastaukset
- kysymyksen kuvaus, jota voidaan käyttää lomakkeessa selventämään kysymyksen tarkoitusta tai haettua vastausta.

Lisäksi followup-sivuille ja peruslomakkeen leesio-osalle syötettiin omat kysymyksen toisiin tauluihin.

5 SÄHKÖISEN LOMAKKEEN OHJELMALLINEN TOTEUTUS

BASE-ACS-lomakkeen toteutuksessa on käytetty ohjelmointikielenä PHP:ta. Se on yksinkertainen ja hyvin yleinen ohjelmointikieli, jolla saadaan HTML-sivuihin lisää toiminnallisuutta. PHP:ta käytettiin juuri siksi, että se on hyvin yleinen ja tulee jatkossakin olemaan yksi tuetuimmista kielistä tähän tarkoitukseen. Suurin osa sivun tiedostoista koostuu PHP-tiedostoista, joissa on sekaisin PHP- ja HTML-kieliä. Lisäksi mukana on yksi CSS-tyylitiedosto ja yksi JavaScript-tiedosto. PHP:n ja MySQL:n versioina käytettiin palvelimen käyttöjärjestelmän mukana tulleita versioita. Nämä versiot ovat yleensä hyvin vakaita ja hyvin testattuja. Suurin osa lomakkeen dokumentoinnista on ohjelman seassa kommenttimuodossa. Varmuuskopiointi suoritettiin manuaalisesti kopioimalla talteen tärkeät tiedostot.

XHTML-merkintäkieli on hyvin tarkka koodin muotoilusta ja kaiken pitää olla niin kuin on määritelty, koska sen tulee toimia mahdollisimman hyvin kaikilla

selaimilla. BASE-ACS-lomakkeen sivut on tarkastettu ulkopuolisella XHTML-verifikaatiosivulla, ja ne noudattavat XHTML Transitional -sääntöjä.

Lomakkeen muodostus alkoi aika yksinkertaisten funktioiden rakentamisesta. Aluksi tarvittiin lähinnä pakolliset komennot HTML-sivun muodostusta varten ja hieman toiminnallisuutta sisällön muodostusta varten. Koodi olisi alusta saakka ehkä kannattanut rakentaa luokkapohjaisesti, jolloin joidenkin toimintojen lisääminen olisi saattanut olla helpompaa, mutta lomake pysyi aika yksinkertaisena loppuun saakka.

Ohjelma koostuu 11 tiedostosta:

- index.php: autentikointi ja XHTML-tiedot
- main.php: päävalikko ja pääfunktioiden kutsu
- patient.php: päälomake
- follow.php: followup-lomake
- update.php: potilaiden tietojen tarkastelu
- main_func.php: suurin osa funktioista
- muut tiedostot: logout.php, csv.php, secure_func.php, style.css ja script.js

Index.php

Tiedostojen hierarkia valuu alaspäin index-tiedostosta, josta kutsutaan aina seuraavaa tarvittaessa PHP:n include-komennolla, joka toimii ainoastaan kun syötetään oikea käyttäjätunnus ja salasana. Käyttäjätietojen tunnistusta varten tarvitaan myös funktiotiedosto, joka sisältää tarvittavat koodit käyttäjätietojen käsittelyä varten.

Index-tiedostossa on XHTML-määrittelyn kannalta tärkeimmät rivit, joissa ilmoitetaan sivun tyyppin määrytykset. Koska sivun oikea sisältö haetaan aina muista tiedostoista include-komennon avulla, näkyvät XHTML-määrytykset aina

jokaisella sivulla. Index-sivulla määritellään myös mitä CSS-tyylitiedostoa ja JavaScript-tiedostoa käytetään.

Main.php-tiedosto on toisena hierarkiassa ja sisältää ohjelman päävalikon. Käyttäjän valinnasta riippuen kutsutaan yhtä viidestä tiedostosta, joihin kuuluvat patient.php, update.php, follow.php, logout.php ja csv.php. Main-tiedosto kutsuu myös pääfunktioitiedoston, joka sisältää suurimman osan ohjelmassa käytettävistä funktioista.

Patient.php-tiedosto on päälomakkeen tulostusta varten. Lomake luetaan kannasta foreach-komennon avulla, joka lukee tietystä taulusta kaikki rivit ja tulostaa ne funktioiden avulla sivulle. Patient-tiedosto on aika pieni rivimäärältään vaikka vaikuttaakin suurelta ominaisuuksiinsa nähden. Suurin osa toiminnallisuudesta on kuitenkin pääfunktioitiedostossa.

Syötettäessä lomakkeeseen tietoa eli kun jossakin lomakkeen kentässä on tietoa ja update-nappulaa painetaan, tieto syötetään kantaan funktioiden avulla. Kantaan luodaan uusi rivi uutta potilasta varten, mutta jos kyseessä on vanhan potilaan tietojen käsittely, päivitetään vanhaan riviin uudet tiedot.

Follow.php-tiedosto on hieman samanlainen kuin Patient.php, koska niiden ominaisuudet ovat lähellä toisiaan. Follow-tiedosto lukee kannasta lomakkeen kysymykset samalla tavalla kuin patient, mutta vain eri taulusta. Follow-tiedosto sisältää myös koodin, jolla listataan kaikki potilaat ja niiden followup-tila.

Update.php on potilaiden listaamiseen käytetty tiedosto, josta pystyy potilaan tunnisteen kanssa siirtymään patient-tiedostoon, jossa potilaan tietoja voi muokata. Ohjelma listaa mm. potilaan tietojen luontipäivämäärän, potilaan numeron ja ryhmän, johon potilas kuuluu.

Main_func.php on tiedosto, joka sisältää suurimman osan lomakkeen toimintaan liittyvistä funktioista. Samalla se on myös helposti suurin tiedosto tässä järjestelmässä. Funktiot ovat vastuussa mm. lomakkeen kysymysten lukemisesta oikeiden otsikoiden alle, kysymyksen tyypistä riippuen oikeiden

HTML-lomakemäärittelyjen tulostamisesta ja lomakkeiden vastausten kirjoittamisesta kantaan.

Muihin tiedostoihin lukeutuvat loput PHP-tiedostot, kuten logout.php, csv.php ja secure_func.php. Logout-tiedosto asettaa sessiotiedot oletusarvoihin eli periaatteessa tyhjentää sessiot, kun päävalikosta valitaan logout. CSV-tiedosto lähettää kaikkien potilaiden tiedot kannasta export.csv-tiedostona. Secure_func-tiedosto sisältää index-tiedostossa tapahtuvan käyttäjän varmistamisen kanssa tarvittavat funktiot.

Script.js-tiedosto sisältää lomakkeen käytössä tarvittavia JavaScript-koodinpätkiä. Tarvitaan lähinnä lomakkeen dynaamisuuden lisäämisessä.

Style.css-tiedosto on pääasiallisesti käytetty tekstien tyylien ja paikkojen määrittelyyn. XHTML-standardissa ei HTML-koodin sisälle laiteta mitään tyylinmäärittelyyn liittyvää ja kaikki hoidetaan ulkopuolisilla määrittelyillä. CSS-tyylit ovat myös hyvä tapa lisätä toiminnallisuutta sivuilla, kuten tässä tapauksessa sivun mukana liikkuva valikko, jossa näkyy myös edistyminen lomakkeen täytössä. (Kuva 4.)

Gender Male Female

Patient number: 1123

Screening	Diseases/History	Pre Procedure	Procedure	Post Procedure	Index Hospitalization / Discharge
2/23	0/20	0/36	0/16	0/36	0/34

Height cm Weight kg

Kuva 4. Sivun mukana liikkuva valikko.

6 TESTAUS JA KÄYTTÖÖNOTTO

Järjestelmän vastaanotto on ollut suurelta osin positiivista. Muutaman erikoisen ongelman takia on testattavaa ollut hieman enemmän kuin aluksi luultiin. Myös eri osapuolten kiireet ja täten vähäinen kommunikointi hieman pitkittivät testausta.

Ensimmäisiä testauksia jarruttivat mm. erikoinen virhe, joka johtui järjestelmän sopimattomuudesta Internet Explorer 6 -selaimella käytettäväksi, joka on ainakin Varsinais-Suomen Sairaanhoidopiirillä toistaiseksi oletusselain. Ilmeisesti IE6 muutti HTML-lomakkeiden koodia lisäämällä lomakkeen HTML-määrittelyihin ylimääräisiä merkkejä. Tätä ei ole vielä enempää tutkittu, koska projektissa eteenpäin pääsemiseksi asennettiin testikoneisiin Firefox-selaimet, joilla järjestelmä toimii normaalisti.

Testaukset ja järjestelmän lopullinen käyttöönotto ovat vielä pilottivaiheessa. Tiedon analysointiin ja CSV-tiedoston sisällön muotoon pitää vielä kiinnittää huomiota. Lopulliseen palvelimeen, jonne järjestelmä siirretään, tulee laittaa kuntoon kunnolliset ajastetut varmuuskopioinnit ja käyttäjien hallinta. Palvelimeksi saattaa tulla jokin yksityisille saatavilla oleva webhotelli.

Tiedon säilyvyyden takaamiseksi voidaan tietokannan sisältö aika ajoin vaikka tulostaa paperille pienellä tekstillä. Automaattiset varmuuskopioinnit useampaan eri kohteeseen ovat myös hyvä tapa ehkäistä tiedon häviäminen yllättävissä tapauksissa.

7 YHTEENVETO

Opinnäytetyössä tutkittiin paperisten ja sähköisten potilastietojenkeruulomakkeiden ominaisuuksia ja verrattiin niitä toisiinsa. Perehdyttiin myös paremmin sähköisen lomakkeen taloudellisiin etuihin ja yleisiin hyötyihin ja haittoihin. Verrattiin myös 4:ä sähköistä lomaketta toisiinsa.

Työssä käytiin läpi miten BASE-ACS-tutkimuksessa käytetty paperinen lomake muutettiin sähköiseksi.

Sähköisen lomakkeen edut paperiseen verrattuna ovat muun muassa: ajan ja kustannusten säästö, lomakkeen käsittelyjen väheneminen ja tilan säästö. Paperisen version hyviin puoliin kuuluu mm. liikuteltavuus ja se, että sen käyttö vaatii vähemmän opettelua. Näistä tuloksista voidaan päätellä, että sähköisiin lomakkeisiin siirtyminen olisi moneltakin osaa kannattavaa, mutta ei välttämättä jokaisessa tapauksessa.

Sähköisten lomakkeiden vertailussa huomattiin, että lomakkeet olivat aika samanlaisia. Lomakkeet olivat ominaisuuksiltaan aika monipuolisia, mutta parhaiten soveltuva lomake saadaan tekemällä sellainen tutkimusta varten.

BASE-ACS-tutkimuksen paperinen kyselylomake koostui n. 25 sivusta kysymyksiä. Nämä kysymykset siirrettiin MySQL-tietokantaan, josta ne pystyttiin PHP-ohjelmalla lukemaan selaimella käytettäväksi sähköisiksi lomakkeiksi.

PHP-ohjelma koostui 11 tiedostosta, jotka oli alusta asti tehty BASE-ACS-tutkimusta varten. Lomakkeiden vastaukset siirrettiin kantaan, josta ne pystyttiin lukemaan selaimella PHP-ohjelman kanssa käsiteltyinä CSV-tiedostoon.

LÄHTEET

- [1] Phoenix Software International, "Case Report Forms", [www-dokumentti]. Saatavilla: <http://www.entrypointplus.com/casereportforms.htm> (Luettu: 8.3.2010)
- [2] Vasankari Tuija, sähköpostikeskustelu, 4.6.2010
- [3] Handelsman David, "Electronic Data Capture: When Will It Replace Paper?", [www-dokumentti]. Saatavilla: <http://www.sas.com/news/feature/hls/sep05edc.html> (Luettu: 29.3.2010)
- [4] Bart Thomas, "Comparison of Electronic Data Capture with Paper Data Collection", [www-dokumentti]. Saatavilla: <http://www.dreamslab.it/doc/eclinica.pdf> (Luettu: 14.4.2010)
- [5] Hyde Andrew William, "The changing face of Electronic Data Capture: from Remote Data Entry to Direct Data Capture", [www-dokumentti]. Saatavilla: <http://organisingchaos.com/Literature/1998.11-DIA.J.pdf> (Luettu: 12.4.2010)
- [6] Clarkston Consulting, "Electronic Data Capture: Expediting Accuracy", [www-dokumentti]. Saatavilla: http://www.clarkstonconsulting.com/viewpoint/doc/Vpt_ElectronicDataCapture_1008.pdf (Luettu: 13.4.2010)
- [7] Hyde Andrew William, "New Technology Systems Being Tried in the Collection of Clinical Trials Data in the Pharmaceutical Industry", [www-dokumentti]. Saatavilla: <http://www.organisingchaos.com/Literature/1998.09-MSc.pdf> (Luettu: 5.6.2010)
- [8] Kuhn T. J., "The difference between eDC and eCRF", [www-dokumentti]. Saatavilla: <http://tjkuhn.wordpress.com/2008/03/14/the-difference-between-edc-and-ecrf/> (Luettu: 24.3.2010)