

NURSEUS -JÄRJESTELMÄN KEHITTÄMINEN

Jukka Pelander

Opinnäytetyö
Toukokuu 2013
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka

TAMPEREEN AMMATTIKORKEAKOULU
Tampere University of Applied Sciences

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka

JUKKA PELANDER:
Nurseus –järjestelmän kehittäminen

Opinnäytetyö 32 sivua, joista liitteitä 2 sivua
Toukokuu 2013

Nurseus, hoitosuunnitelmien ja yhteenvedojen kirjaamisalusta, kehitettiin opetuskäyttöön TAMK:n sairaanhoitajaopiskelijoita varten. Järjestelmään kirjaamisen perustana on kansallinen FinCC-luokitusjärjestelmä, joka löytyy terveyden ja hyvinvoinnin laitoksen Internet-sivuilta. Hoitotyön päivittäiseen kirjaamiseen tarkoitettu FinCC-luokituskokonaisuus muodostuu Suomalaisesta hoidon tarveluokituksesta (SHTaL), Suomalaisesta hoitotyön toimintoluokituksesta (SHToL) sekä hoidon tuloksen luokituksesta (SHTuL). Ohjelmaan on otettu hieman mallia työelämän käytössä olevista järjestelmistä.

Järjestelmän ensimmäisessä kehitysvaiheessa uusittiin järjestelmän tekniikka lähes kokonaan. Tietokanta koki myös suuria muutoksia. Tietokantamoottori vaihdettiin, jotta saatiin viite-eheys kuntoon tietokannassa. Lisäksi taulut järjesteltiin uudelleen, sillä saatiin kaikki tieto tallentumaan tietokantaan ja luovuttiin XML-tallennuksista. Käyttäjää ajateltiin lisäämällä käyttömukavuutta AJAX-komponenteilla ja tarkastuksilla. Uutena komponenttina järjestelmään tehtiin hoitosuunnitelmista tehtävä yhteenvedo. Yhteenvedossa tiivistetään hoitosuunnitelma FinCC-luokituskokonaisuuden komponenttitasolla.

Käytössä oleva järjestelmä vaatii kehittämistä jatkossakin. Käyttäjät antoivat järjestelmästä sähköisesti palautetta nimellä tai nimettömästi. Saatu palaute analysoitiin ja käyttäjäpalautteen perusteella kirjattiin jatkokehitysehdotukset. Ulkoasun päivitys koettiin tarpeelliseksi uudistukseksi ja sen tekijäksi tarvitaan käyttöliittymäsuunnittelija, graafikko. Tekniikan päivittämiseen ja joidenkin ominaisuuksien päivittämiseen tarvitaan tietotekninen henkilö, ohjelmistotekniikan alalta.

Asiasanat: nurseus, mysql, php, ajax

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree programme in Computer Science
Software Engineering

JUKKA PELANDER:
The development of Nurseus

Bachelor's thesis 32 pages, appendices 2 pages
May 2013

Nurseus was developed for educational use for the nurse students of TAMK. The system is based on the national FinCC rating system, which can be found in the National Institute for Health and Welfare (THL) Web site. The program is modeled on the system at work.

System underwent major changes in the first stage of development. Database engine was changed that referential integrity was fixed in the database. Database tables also were restructured. All the data is now stored in the database and the same time was removed from storage in XML-format. Usability was added to AJAX components. The system was implemented as a new component, summary made from the treatment plan. In the summary the data from the treatment plan is summarized based on the FinCC rating system on component level.

The system was collected feedback. Feedback was given anonymously. The feedback was analyzed and further developments of the ideas were documented. The necessary works of updating of layout and some technology of the system were recorded.

Key words: nurseus, mysql, php, ajax

SISÄLLYS

1	JOHDANTO.....	7
2	ALKUPERÄINEN NURSEUS-JÄRJESTELMÄ	8
3	OHJELMISTOTUOTANTO / -KEHITYS	10
3.1	Käytetyimpiä menetelmiä	10
3.1.1	Vesiputousmalli	10
3.1.2	V-malli	12
3.1.3	Ketterät menetelmät	12
4	KÄYTETYT MENETELMÄT	14
5	KÄYTETYT TEKNIIKAT JA TYÖKALUT	15
5.1	PHP	15
5.2	MySQL	16
5.2.1	SQL-injektio.....	17
5.3	JavaScript.....	18
5.4	AJAX	18
5.5	CSS	18
5.6	Notepad++	19
5.7	MySQL workbench.....	19
5.8	PuTTY	19
6	PÄIVITETTY JÄRJESTELMÄ.....	21
6.1	Hoitosuunnitelma.....	21
6.2	Yhteenveto	24
6.3	Muita muutoksia	26
7	KÄYTTÖÖNOTTO JA OHJEISTUS.....	27
8	YHTEENVETO JA JATKOKEHITYS	28
	LÄHTEET.....	29
	LIITTEET	31
	Liite 1. Järjestelmän ER-kaavio	31
	Liite 2. Tilasto Nurseuksesta	32

LYHENTEET JA TERMIT

AJAX	Asynchronous JavaScript and XML
ACID	Atomicity, Consistency, Isolation, Durability - periaate, jolla turvataan järjestelmän tietojen eheys kaikissa tilanteissa
bugi	ohjelmointivirhe
C++	yksi tärkeimmistä kaupallisessa ohjelmistokehityksessä käytettävistä ohjelmointikielistä
Chrome	Googlen kehittämä selainohjelma, joka on saatavilla Windowsille, Linuxille, Mac OS X:lle
CSS	Cascading Style Sheets
ER-mallinnus	Entity Relationship-modeling - oliopohjainen menetelmä, jolla voidaan kuvata käsitteitä ja niiden välisiä suhteita
Firefox	avoimen lähdekoodin www-selain, joka on saatavilla Windowsille, Linuxille, Mac OS X:lle
Git	versionhallintajärjestelmä
GPL	lisenssi, joka antaa kenelle tahansa oikeuden käyttää, kopioida, muuttaa ja jakaa edelleen ohjelmia ja niiden lähdekoodia
HTML	Hypertext Markup Language - avoimesti standardoitu kuvauskieli, jolla voidaan kuvata hyperlinkkejä sisältävää tekstiä
hyperlinkki	Hyperlink - kuva, teksti tai sana, joka siirtää käyttäjän sivulta toiselle webissä
InnoDB	MySQL:n tietokantamoottori, joka tukee vierasavaimia (foreign key) sekä transaktioita
Java	Sun Microsystemsin kehittämä laaja teknologiaperhe ja ohjelmistoalusta, johon kuuluu muun muassa laitteistoriippumaton oliopohjainen ohjelmointikieli sekä ajoaikainen ympäristö virtuaalikoneineen ja luokkakirjastoineen
JSP	JavaServer Pages - Javaan perustuva menetelmä luoda HTML- ja XML muotoisia websivuja
Linux	viittaa Linux-ydintä käyttävien Unixin kaltaisten käyttöjärjestelmien perheeseen
Mac Os	viittaa Applen Macintosh-tietokoneisiinsa kehittämien käyttöjärjestelmien sarjaan

mm.	muun muassa
MVC	model-view-controller - ohjelmistoarkkitehtuurityyli, tarkoituksena on käyttöliittymän erottaminen sovellusalueesta.
MySQL	Avoimen lähdekoodin relaatiotietokanta
op	opintopiste
PDO	PHP Data Objects - PHP 5.1:n mukana tullut uusi ominaisuus tietokantojen käsittelyyn.
PHP	ohjelmointikieli, jota käytetään erityisesti Web-palvelinympäristöissä dynaamisten web-sivujen luonnissa
skripti	komentosarja
SCRUM	ketterän ohjelmistokehityksen menetelmä
sprint	SCRUM menetelmän osasuoritus
SQL	Structured Query Language – standardoitu relaatiotietokannan kyselykieli
SQL-injektio	SQL injection - tekniikka tietoturva-aukkojen hyödyntämiseen tietokantapohjaisissa sovelluksissa
SQLite	relaatiotietokantajärjestelmä
SSH	Secure Shell - salatun tietoliikenteen protokolla
STL	Standard Template Library
svn	Subversion - versionhallintajärjestelmä
TAMK	Tampereen ammattikorkeakoulu
transaktio	transaktiolla varmistetaan, että kaikki haluttu tieto lisätään tietokantaan ja transaktion jäljiltä tietokannan tulee olla johdonmukaisessa tilassa ja sen pitää säilyä mahdollisen kaatumisenkin jälkeen
vierasavain	Foreign Key (FK) – luo yhteyden toisen taulun kanssa, vierasavain viittaa toisen taulun pääavaimeen
viite-eheys	määrittävät mitä yhdistetyille tiedoille tapahtuu jos taulujen tiedot vaihtuvat tai ne poistetaan
web-sovellus	internetin kautta jaeltava ohjelmisto
Windows	Microsoft:n kehittämä PC:lle tarkoitettujen graafisten käyttöliittymien ja käyttöjärjestelmien perhe, joka esiteltiin vuonna 1985
XML	Extensible Markup Language - rakenteellinen kuvauskieli, joka auttaa jäsentämään laajoja tietomassoja selkeämmin

1 JOHDANTO

Tampereen ammattikorkeakoulussa on käytössä ainutlaatuinen Nurseus -niminen hoitosuunnitelmien kirjaamisalusta, johon opiskelijat kirjaavat kuvitteellisten case-potilaiden (harjoituspotilas) tai harjoittelussa hoitamiensa oikeiden potilaiden hoitosuunnitelmia. Hoitosuunnitelmien lisäksi järjestelmään tehtiin uutena ominaisuutena hoitosuunnitelmasta tehtävä yhteenveto. Järjestelmä on suunniteltu opiskelijoiden harjoitustöiden tekemiseen, joita opettajat voivat käydä tarkastelemassa ja arvioimassa. Nurseuksen päätehtävä on tutustuttaa opiskelijat FinCC -luokituksiin, joita he työelämässä tarvitsevat tehdessään potilaista virallisia hoitokertomuksia.

Tämä opinnäytetyö keskittyy Tampereen ammattikorkeakoulun sosiaali- ja terveysalan opiskelijoiden käytössä olleen Nurseus -järjestelmän jatkokehitykseen. Tärkeimpinä kriteereinä oli saada ohjelmasta helppokäyttöisempi uusien tekniikoiden avulla ja karsia päällekkäisiä toimintoja helpommin hallittaviksi kokonaisuuksiksi.

Työssä käsitellään aluksi alkuperäistä järjestelmää ja sen puutteita ja kehityskohteita. Luvussa kolme kerrotaan ohjelmistokehityksestä yleisesti. Luvussa viisi käydään läpi tekniikoita, joita käytettiin työn tekemiseen. Samassa luvussa esitellään myös työvälineet, jotka mahdollistivat työn tekemisen. Tekniikoiden jälkeen, luvussa kuusi, on katsaus päivitetystä Nurseus -järjestelmästä. Seitsemännessä luvussa käsitellään järjestelmän käyttöönottamista ja järjestettyjä koulutustilaisuuksia. Työn lopuksi, luvussa kahdeksan, pohditaan järjestelmän jatkokehitystä.

2 ALKUPERÄINEN NURSEUS-JÄRJESTELMÄ

Nurseus –järjestelmä on opiskelijoiden harjoituskäyttöön kehitetty web-sovellus. Ohjelma koostui alun perin ainoastaan hoitosuunnitelman kirjaamiskomponentista, joka vaati käyttäjäkokemuksen perusteella kehittämistä. Kirjaamisen perustana on kansallinen FinCC –luokitusjärjestelmä. Alkuperäisessä järjestelmässä oli myös käytetty CSS -tyylitiedostoa, joten päätettiin jatkaa samalla hyväksi havaitulla linjalla.

Hoitosuunnitelma tehdään siinä vaiheessa, kun potilas saapuu hoitoon. Hoitosuunnitelmaan kirjataan potilaan antamat esitiedot (Nurseus-järjestelmässä: potilaan perustiedot), potilaan kertomat oireet hoitoon tullessa (tarve), hoitaja tekemä tavoite potilaan näkökulmasta (tavoite), potilaalle tehtävät toimenpiteet (suunnitellut toiminnot) ja potilaalle tehdyt toimenpiteet (toteutus).

Kyseessä on kolmitasoinen järjestelmä, jossa on komponentti-, pää- ja alaluokat. Komponenttitaso oli vähintään valittava. Järjestelmään kirjattiin hoidon tarve ja toiminto valiten ensin komponenttitaso, jonka jälkeen siirryttiin seuraavaan valintaan ja valittiin mahdollinen pääluokka ja sen jälkeen samalla tavalla mahdollinen alaluokka, jos pääluokka sisälsi alaluokkia. Virheen sattuessa koko polku oli aloitettava alusta. Tähän kohtaan haluttiin parannuksia aikaan, joten se oli tärkein kehitettävä ominaisuus järjestelmässä.

Alkuperäisen Nurseus-järjestelmän tietokanta paljastui myös puutteelliseksi. Kehitystyön alussa huomattiin järjestelmän ER-kaaviosta katsottuna, että tietokannassa ei ole ainuttakaan viiteavainta eli taulujen välillä ei ole mitään yhteyksiä. Kaaviossa oli erilliset taulut kirjautumistiedoille, FinCC –luokituksille, työn perustiedoille ja arviointitiedoille mutta töiden sisältö puuttui kokonaan. Töiden sisältö oli tallennettu palvelimen kovalevyllä XML-muodossa. Näiden tietojen perusteella järjestelmästä piti saada helpommin hallittava eli tietokanta täytyi kehitystyön aluksi suunnitella uudelleen. Käytössä olevan tietokannan korjaus oli teknisesti suurin ja haastavin kokonaisuus kehitystyössä. Kehitys tietokantaan oli myös tärkeää, että järjestelmästä saatiin eheä ja kaikki tieto saatiin järjesteltyä samaan paikkaan.

Järjestelmään haluttiin uutena komponenttina yhteenveto. Yhteenvedosta oli paperiver-
sio, joka haluttiin kehittää sähköiseen muotoon hoitosuunnitelman kirjaamisen rinnalle.
Järjestelmään haluttiin myös määräaikaista käyttäjiä, mikäli se olisi mahdollista.

Opettajilta kerätyn palautteen perusteella työn arviointi kohtaa oli kehitettävä helppo-
käyttöisemmäksi. Alkuperäisessä järjestelmässä työ avattiin uuteen ikkunaan tarkastelua
varten ja toiseen ikkunaan avattiin arviointilomake. Opettajanäkymän töiden ja arvioitu-
jen töiden selaus näkymät haluttiin päivittää. Pääkäyttäjän kehityskohteista tärkeimmäk-
si kirjattiin uusien FinCC –luokitusten vieminen tietokantaan. Alkuperäisessä järjestel-
mässä luokituksia pystyi muokkaamaan/lisäämään yksitellen.

3 OHJELMISTOTUOTANTO / -KEHITYS

Ohjelmistotuotanto on yhteisnimitys niille työnteon ja työnjohdon menetelmille, joita käytetään, kun tuotetaan tietokoneohjelmia. Laajasti ymmärrettyinä ohjelmistotuotanto kattaa kaiken tietokoneohjelmistojen valmistukseen liittyvän prosessinhallinnan sekä kaikki erilaiset tietokoneohjelmien valmistamisen menetelmät. Tarve valmistukseen tulee useimmiten asiakkaalta tai ohjelman tuottaja tekee markkinointitutkimuksen ja pääättelee tarpeen ohjelmistolle. (Haikala & Mikkonen, 2011)

Ohjelmistotuotannossa tieteenhaarana tutkitaan myös ohjelmien rakenteellisia ominaisuuksia kuten dokumentointia, versionhallintaa, sekä jäljitettävyyttä. Ohjelmistotuotantoon kuuluvat myös erilaiset laatu järjestelmät, joita käytetään erityisesti yrityksissä dokumentoimaan yrityksen toimintatapoja. Laatu järjestelmän tavoitteena on dokumentoida ja ottaa käyttöön hyväksi havaitut toimintatavat, jotka parantavat yrityksen toimintaprosessin laatua. (Haikala & Mikkonen, 2011)

Kaupallista ohjelmistokehitystä tehdään yleensä projektityönä. Toisaalta useiden ohjelmistojen kehitys ja ylläpito jatkuvat koko niiden elinkaaren ajan ilman ennakoitavaa päättepistettä. (Haikala & Mikkonen, 2011)

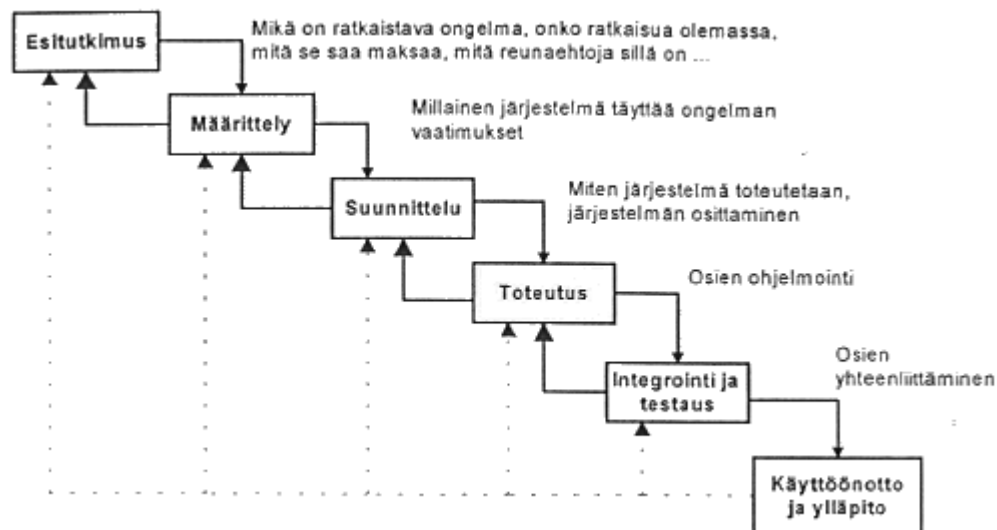
3.1 Käytetyimpiä menetelmiä

Ohjelmiston tekeminen on usein haastava tehtävä ja se sisältää erilaisia vaiheita. Vaihejakomalliksi kutsutaan ohjelmiston suunnittelua siten, että ohjelman kehitys tai koko elinkaari jaetaan sopivan kokosiin osiin. Vaihejakomalleja ovat mm. vesiputousmalli, prototyypimalli ja ketterät menetelmät. Perinteisin malli näistä on vesiputousmalli (waterfall model) ja nykyaikaisin on ketterät menetelmät (agile software development).

3.1.1 Vesiputousmalli

Vesiputousmalli (kuva 1) on vanhin mutta samalla yleisin käytössä oleva vaihejako malli. Sen kehitys alkoi 1960-luvulla, kun ensimmäisten ohjelmistojen valmistusprojek-

tissa tarvittiin systemaattisia työprosesseja. Vesiputousmallin rakenne muistuttaa vesiputousta, jossa vaiheet seuraavat toisiaan tietyssä järjestyksessä. Puhtaassa vesiputousmallissa uusi vaihe aloitetaan edellisen päätyttyä. Ajatuksena on, että kukin vaiheista tuottaa dokumentin tai joukon dokumentteja, jotka toimivat syötteenä seuraavalle vaiheelle. Taaksepäin peruuttamista pyritään välttämään tekemällä jokainen vaihe kerralla loppuun saakka, jotta tarvetta palaamiseen ei tulisi. (Haikala & Mikkonen, 2011)



KUVA 1. Vesiputousmalli (Immonen, 2003)

Koska vesiputousmalli on osoittautunut vaikeaksi noudattaa sellaisenaan, sitä käytetään ohjelmistoprojektissa usein soveltaen. (Haikala & Mikkonen, 2011)

Vesiputousmalli perustuu hyvään dokumentaatioon ja dokumenttien huolelliseen tarkastamiseen. Dokumentin tarkistamisen jälkeen se jäädytetään, eikä sitä voi enää muokata ilman projektijohdon lupaa. Tämä tarkoittaa sitä, että uudet vaatimukset suunnitellaan vasta seuraavaan ohjelmaversioon. Tässä mallissa asiakas saa valmiin tuotteen nähtäväksi vasta projektin päättyessä. Nämä tekevät vesiputousmallista hieman kankean nykyaikaiseen ohjelmistokehitykseen. Vesiputousmallin etuina on, että malli on selkeä ja se on helposti omaksuttavissa. Lisäksi mallilla on laaja työkalu- ja teorialuki. Kiireiset asiakkaat suosivat tätä mallia, sillä heidän ei tarvitse projektin edetessä osallistua projektityöhön kuten ketterissä menetelmissä.

3.1.2 V-malli

V-malli on hieman joustavampi ja kehittyneempi lineaarinen malli vesiputousmallista. V-mallissa jokaista suunnitteluvaihetta vastaa testausvaihe (kuva 2). Testitapaukset määritellään suunnittelun ja toteutuksen kanssa samaan aikaan.



KUVA 2. V-malli

V-mallissa on tärkeää myös huolellinen dokumentaatio mutta dokumenttien tarkistus ei ole yhtä tarkkaa kuin vesiputousmallissa. Puhtaassa V-mallissa suunnitelma ja vastaava testitapahtuma tarkastetaan samaan aikaan.

3.1.3 Ketterät menetelmät

Ketterä ohjelmistokehitys (agile software development) on joukko ohjelmistotuotanto-projekteissa käytettäviä menetelmiä, joille on yhteistä toimivan ohjelmiston ensisijaisuus, suora viestintä ja nopea muutoksiin reagointi. Ketteriä menetelmiä on useita, mm. Extreme Programming (XP), Scrum, DSDM, Crystal Methods, Agile modeling, Adaptive software development, Pragmatic Programming, Feature driven development ja Gilb-EVO. (Wikipedia, ketterä ohjelmistokehitys, 2013)

Ketterissä menetelmissä ohjelmistoa kehitetään tiiviissä yhteistyössä asiakkaan kanssa. Tässä mallissa vaatimukset voivat muuttua projektin edetessä paljonkin eikä se estä projektin suorittamista loppuun, kuten esimerkiksi vesiputousmallissa. Kaikkien ketterien menetelmien yhteisiin piirteisiin kuuluu riskien minimointi jakamalla kehitystyö pie-

niin osiin, jotka tyypillisesti kestävät viikosta neljään viikkoon. Scrum:ssa näitä osia kutsutaan nimellä sprint. Yksittäinen sprint koostuu suunnittelusta, toteutuksesta, testauksesta ja dokumentoinnista. Seuraavan sprintin sisältö päätetään aina edellisen sprintin päättyessä. Jokaisen sprintin jälkeen järjestelmästä on oltava toimiva versio, jonka voi esitellä asiakkaalle.

4 KÄYTETYT MENETELMÄT

Tätä työtä tehtäessä käytettiin sovelletusti montaa eri mallia. Koska kyseessä oli yhden kehittäjän projekti, niin tämä onnistui. Aluksi käytössä oli sovellettu vesiputousmalli, koska ohjaavat opettajat olivat lomalla sen ajan (kesä-heinäkuu) kun ohjelmaa varsinaisesti kehitettiin. Ohjaajat antoivat kesäksi ohjeet, määritykset ja vaatimukset. Projektin lopussa (elokuu), ennen järjestelmän käyttöönottoa, kehitystyössä sovellettiin ketteriä menetelmiä, jotta pienetkin yksityiskohdat saatiin kohdilleen ennen järjestelmän julkaisua.

Työssä sovellettiin myös suihkulähdemallia. Suihkulähdemalli on ohjelmistotuotantomalli, jossa ohjelmointi suoritetaan ensin, dokumentointi vasta sen jälkeen. Suihkulähdemallissa tuotannon järjestys on siis päinvastainen kuin vesiputousmallissa. (Wikipedia, suihkulähdemalli, 2012)

Projektin alussa järjestelmä oli vielä ahkerassa käytössä opiskelijoilla, joten päätettiin järjestelmästä irrottaa erillinen kehitysympäristö, jossa kehitystyö tapahtuu ilman, että se vaikuttaa tuotannossa olevan järjestelmän toimintaan. Kehitysympäristöön päädyttiin myös siitä syystä, jos kehitystyö epäonnistuu tai ei ole halutun kaltainen, voidaan jatkaa vanhan version käyttöä. Kehitysympäristöön toteutettiin kaikki muutokset, järjestelmän toiminnallisuuden kehittäminen ja halutut uudet komponentit. Ohjaajat pääsivät kirjautumaan kehitysympäristöön koko projektin ajan ja pystyivät seuraamaan projektin edistymistä. Projektin lopussa kehitysympäristö käytiin läpi ohjaajien kanssa. Muutokset hyväksyttiin ja järjestelmä siirrettiin tuotantokäyttöön.

Varsinaista kehitystyötä tehtäessä valittiin, että varsinaista versionhallintaa (svn, git) ei käytetä, koska kyseessä on yhden kehittäjän projekti, eikä projektin suuruus sitä välttämättä edellytä. Tässä otettiin se riski, että ei voida helposti palata toimivaan versioon, jos ohjelmakoodiin tulee myöhemmin koko järjestelmän hajottava ohjelmointivirhe, jonka korjaaminen vie paljon aikaa.

5 KÄYTETYT TEKNIIKAT JA TYÖKALUT

5.1 PHP

PHP on laajasti käytetty yleiskäyttöinen skriptikieli, joka sopii erityisesti Web-kehitykseen ja joka voidaan sulauttaa HTML:n sekaan. PHP:ssa on vaikutteita ohjelmointikielistä: Perl, C, C++, Java ja Tcl. (PHP, 2013)

PHP työkalut on julkaistu GPL-lisenssillä vuonna 1995 ja PHP on peräisin alun perin sanoista: "Personal Home Page" (Wikipedia, PHP, 2013)

PHP-koodi kirjoitetaan tagien "<?php" ja "?>" väliin ja useimmiten se upotetaan HTML-tagien joukkoon. PHP on heikosti tyyhitetty kieli eli muuttujan tyyppi määräytyy sijoitettavan arvon perusteella. Muuttujat alkavat dollarimerkillä \$, esimerkiksi \$palkka. PHP -koodi käännetään PHP-tulkilla. Tulkki palauttaa käännetyn koodin palvelimelle, josta se lähetetään edelleen käyttäjän selaimeen.

TAULUKKO 1. PHP:n historiaa (Wikipedia, PHP, 2013)

Versio	Julkaisuvuosi	Huomioita
1.0	1995	Julkaisu
2.0	1997	"nopein ja yksinkertaisin työkalu" luoda dynaamisia Web sivuja
3.0	1998	Lähdekoodin kokonaan uudelleenkirjoitus
4.0	2000	Uusi ydin, tuki kolmansien osapuolten ohjelmistorajapinnoille
5.0	2004	Olio-ohjelmointituki, sisäänrakennettu tietokantamoottori (SQLite)
5.4.12	2013	Uusin versio

Alkuperäinen järjestelmä oli kirjoitettu PHP:llä. Se edesauttoi, että PHP valittiin myös jatkokehitykseen. PHP oli myös tuttu kieli kehittäjälle ja aikaisempi kokemus vahvistivat valintaa. Vaihtoehtoina olisi ollut myös muita kehitysympäristöjä, esimerkiksi Spring. Kokemukset Spring -työkaluista eivät olleet kovin hyviä, joten se hylättiin. Spring perustuu MVC -arkkitehtuurimalliin, Java -ohjelmointikieleen ja JSP -sivujen

käyttöön. Koska kehitysaika ja –resurssit olivat rajalliset ja työtä paljon, päädyttiin käyttämään kehittäjän vahvemmin osaamaa tekniikkaa. Näin myös varmistettiin, että järjestelmä on valmis julkaistavaksi projektin päättyessä.

PHP:llä on helppo omaksua ja sen avulla saa helposti ja melko vaivattomasti web-sivut aikaiseksi. Tekniikka on helppo ja koodin voi upottaa suoraan HTML:n sekaan ja perus sivusto on nopeasti valmis. Spring framework vaatii jo aluksi paljon enemmän erilaisia määrittelyjä, jotta sivut saadaan näkymään oikein. Tämä vie paljon aikaa itse kehitykseltä. Spring:n etu puolestaan on, että näkymät on erotettu kokonaan sovelluslogiikasta mvc –arkkitehtuurilla eli logiikkaa ei tarvitse tehdä joka näkymälle uudestaan ja uusien näkymien tekeminen on helppoa myös jälkikäteen.

Tässä projektissa käytetään tietokannan käsittelyssä PDO –tietokantaohjelmointimallia, joka se on ollut PHP:n versiosta 5.1 saakka käytössä. PDO on joukko luokkia, joiden avulla tietokantaa käsitellään. Lisäksi PDO on joustava, koska sitä voi käyttää lähes minkä tietokannan kanssa tahansa, kunhan sille on olemassa PDO-ajuri. PDO:n käyttöön päädyttiin, koska se helpottaa tietokannan käsittelyä ja vähentää työmäärää. Työmäärän väheneminen johtuu siitä, että itse kirjoitettavien tarkastusten määrä vähenee, koska PDO:n luokat tekevät ne automaattisesti. Tämä perustuu siihen, että SQL-kyselyt valmistellaan aluksi oikean muotoiseksi, jonka jälkeen halutut parametrit sijoitetaan SQL-lausekkeen haluttuun kohtaan. Samalla PDO siis lisää järjestelmän turvallisuutta ja poistaa mm. SQL-injektion mahdollisuuden kokonaan.

5.2 MySQL

MySQL on vuonna 1995 kehitetty avoimen lähdekoodin relaatiotietokanta ja se oli käytetyin tietokanta vuonna 2008. MySQL on kirjoitettu ohjelmointikielillä C ja C++ ja se perustuu SQL:ään. SQL on IBM:n kehittämä standardoitu tietokannan kyselykieli, jolla tehdään haluttuja kyselyitä, muutoksia ja lisäyksiä. MySQL–tietokantaa käyttävät myös tunnetut palvelut: Wikipedia, Google, Facebook, Twitter, Flickr, Nokia.com ja YouTube. Sun Microsystems osti MySQL:n kehittäneen yrityksen 2008 ja ohjelmistoyritys Oracle Corporation osti Sun Microsystemsin 2009. Kaupan yhteydessä MySQL:n omistus siirtyi Oraclelle. (MySQL, 2013) Oraclella on myös oma kaupallinen tietokanta. Tietokantaa voi käyttää komentoriviltä tai jollain graafisella työkalulla, esimerkiksi

phpMyAdmin. Komentorivikomennolla muutos tapahtuu suoraan tietokantaan ilman tarkistuksia, jos syötetty syntaksi on oikein ja graafisella ohjelmalla pääsee välillisesti muokkaamaan tietokantaa.

MySQL valittiin kehittäjän mieltymysten perusteella ja koska alkuperäisessä järjestelmässä oli pohjana myös MySQL -tietokanta. Vaihtoehtona olivat myös avoimen lähdekoodin tietokannat: PostgreSQL oliotietokanta ja uusi MariaDB. Uusi MariaDB-tietokanta hylättiin ajanpuutteen, kokemattomuuden ja heikosti saatavilla olleiden dokumenttien puuttumisen vuoksi. Oliopohjainen tietokanta PostgreSQL olisi vaatinut myös paljon tutustumista ennen kehitystyön aloittamista, niin se hylättiin. Saattoi valintaa vaikuttaa kehittäjän myönteinen kokemus MySQL:stä helppona ja nopeana tietokantana.

Tietokannan täytyi tukea tärkeimpiä ACID-ominaisuuksia eli viite-eheyttä (vierasavaimet) ja transaktioita. Nämä ovat relaatiotietokannassa tiedon hallittavuuden kannalta lähes välttämättömiä. Viite-eheydellä tarkoitetaan, että vierasavaimilla viitataan toisen taulun pääavaimeseen. Näin kanta pysyy ehjänä eli eheästä tilasta siirrytään aina toiseen eheään tilaan. Transaktiolla tarkoitetaan joukkoa tapahtumia, jotka suoritetaan peräkkäin siten, että kaikkien joukkoon kuuluvien tapahtumien on onnistuttava ennen kuin transaktio hyväksytään suoritetuksi, muulloin kaikki tapahtumat perutaan. Eli lyhyesti transaktiossa suoritetaan kaikki siihen kuuluvat tapahtumat tai ei mitään.

Tietokannan suunnittelussa käytettiin apuna MySQL Workbench -ohjelmaa. Ohjelmalla pystytään suunnittelemaan tietokannan rakenne ER -kaavion avulla. Nurseus -järjestelmän ER -kaavio on tämän dokumentin liitteenä (liite x). MySQL Workbench -ohjelmalla voi ottaa myös yhteyden tietokantaan ja suorittaa komentosarjan, jolla ER -kaavio muutetaan SQL-komennoiksi ja ne ajetaan tietokantaan eli luodaan tietokantataulut ja lisätään oletusdata tietokantaan.

5.2.1 SQL-injektio

Tietokantapohjaisissa sovelluksissa esiintyy ohjelmointi virheestä johtuvaa SQL-injektiota. Tämä oli vuonna 2011 yleisin ohjelmointivirhe pistein 93,8/100. (CWE/SANS Top 25 Software Errors, 2011) SQL-injektiossa on kyse siitä, että käyttä-

jän antamaa syötettä ei tarkasteta vaan se syötetään suoraan sellaisenaan tietokantaan. Jos käyttäjän syötettä ei tarkasteta, voi käyttäjä syöttää omia SQL-lausekkeitaan tietokantaan niin, että kukaan muu ei sitä välttämättä huomaa. Näin järjestelmästä saa helposti kirjautumistiedot tulostettua tai niin halutessaan tietokannan saa ajettua alas ja kaikki tiedot poistettua. Tämä haavoittuvuus on hyvin hyökkääjien tiedossa mutta onneksi tämä haavoittuvuus on todella helppo ja halpa korjata. Käyttäjän antama syöte tulee aina tarkastaa.

5.3 JavaScript

JavaScript on alun perin Netscape Communications Corporationin kehittämä pääasiassa Web-ympäristössä käytettävä skriptikieli eli komentosarjakieli. JavaScriptiä käytetään miljardeilla web-sivuilla. JavaScriptin tärkein sovellus on mahdollisuus lisätä Web-sivuille dynaamista toiminnallisuutta, lomakkeiden tarkastuksia, palvelimen kanssa kommunikointia ja paljon muuta. JavaScriptiä ei tule sekoittaa Javaan, ne ovat kaksi erilaista kieltä. (W3School - JavaScript)

5.4 AJAX

Käyttäjäystävällisyyden lisäämiseksi valittiin AJAX tekniikka, jotta sivuja ei kokonaan tarvitse päivittää joka vaiheessa. Järjestelmän sivuja pystytään päivittämään AJAX:n avulla niin, että koko sivua ei tarvitse ladata uudelleen. Tekniikka perustuu lyhyesti siihen, että toimintoja käydään suorittamassa erillisellä, esimerkiksi .php, sivulla. Erillisille sivuille kutsut hoidetaan JavaScript-kommenteilla, joilla AJAX:n avulla, toiselta sivulta, saatu tulos sijoitetaan kutsun lähettäneellä sivulla haluttuun kohtaan.

5.5 CSS

CSS on tyyli tiedosto, jolla voidaan hallita web-sivuston tyyliä ja ulkoasua samasta paikasta. Tyyli tiedoston käyttäminen helpottaa php-koodin ja html-tagien lukemista, koska kooditiedostoissa on vain viite, mistä tämän kohdan tyyli löytyy, eikä tyylimääriä

ole tungettu varsinaisen lähdekoodin sekaan. Jälkeenpäin ulkoasun päivittäminen on helpompaa järjestelmässä, jossa tyylit ovat kaikki samassa tiedostossa.

5.6 Notepad++

Notepad++ on avoimenlähdekoodin tekstieditori Windows-ympäristöön. Notepad++ on kirjoitettu C++:lla ja se käyttää Windowsin ohjelmointirajapintaa ja STL:ää. Nämä takaavat korkean suorituskyvyn ja pienemmän ohjelman koon. Tuotteella on GPL-lisenssi ja se on käännetty monelle kielelle. (Notepad++ Home)

Tämä tekstieditori valittiin projektiin hyvien kokemusten vuoksi. Editori on hyvä työkalu koodaajalle, koska siinä on värikoodaustuki monelle eri ohjelmointi-, skripti- ja merkintäkielen syntaksille, esimerkiksi php, C++, java, C#, python, mikä helpottaa kehittäjän työtä.

5.7 MySQL workbench

MySQL Workbench on visuaalinen työkalu tietokanta-arkkitehtuurin, kehittämisen tekemiseen. MySQL Workbench tarjoaa data mallinnusta, SQL kehitystä ja kattavat hallintatyökalut palvelimen kokoonpanon, käyttäjähallintaan, ja paljon enemmän. MySQL Workbench on saatavana seuraaville alustoille: Windows, Linux ja Mac OS. (MySQL Workbench 5.2)

Ohjelmalla pystytään mallintamaan tietokanta ER-kaavion avulla. ER-kaavio sisältää tiedot tietokannan tauluista ja niiden välisistä yhteyksistä, viite-eheys. Kaavion pystytään määrittämään myös taulujen sisältö. Kaavion valmistuttua se pystytään siirtämään suoraan ohjelmasta halutulle palvelimelle. Ohjelma muuttaa tässä vaiheessa ER-kaavion SQL-komennoiksi, jotka ajetaan tietokantaan. Tämä edellyttää, että tietokantaan on tunnuksilla oikeudet ja tunnuksella ainakin oikeudet lisätä tietokantaan tietoa.

5.8 PuTTY

PuTTY on SSH ja telnet asiakasohjelmisto. Se kehitettiin alun perin Windows-alustalle. PuTTY on avoimen lähdekoodin ohjelmisto, jonka lähdekoodin on saatavilla. Joukko vapaaehtoisia tukee ja kehittää PuTTY -ohjelmaa. (putty.org) PuTTY-ohjelmaa käytettiin tässä projektissa tietokannan suoraan muokkaamiseen komentoriviltä käsin.

6 PÄIVITETTY JÄRJESTELMÄ

Tämä luku käsittelee Nurseus-järjestelmän sisältöä. Luvussa kerrotaan myös järjestelmään tehdyistä muutoksista. Järjestelmän tärkeimmät komponentit ovat hoitosuunnitelman ja yhteenvedon kirjaaminen ja niitä käsitellään seuraavaksi. Etusivua selkeytettiin poistamalla epäolennaiset tiedot (kuva 3).

Actions

Etusivu

[Muokkaa omia tietoja](#)

[Ryhmähallinta](#)

[Työt](#)

[Ohjeet](#)

Anna järjestelmästä palautetta:

[Palaute](#)

Etusivu

Huomioitavaa:

Sivusto toimii parhaiten selaimilla: [Mozilla Firefox](#) ja [Google Chrome](#)

Viimeisin arviointi: **26.4.2013 08:56:00**

Olet palauttanut hoitosuunnitelmat:

Testiryhmä: harjoitustyö_kokkora (Tarkastettu)

Testiryhmä: harjoitustyö_juonka (Tarkastettu)

Testiryhmä: Harjoitustyö_t8jpelan (Tarkastettu)

Testiryhmä: Harjoitustyö_vesalhe (Tarkastettu)

Testiryhmä: harjoitustyö_seitssu (Tarkastettu)

Testiryhmä: Harjoitustyö_nikida (Tarkastettu)

Testiryhmä: Harjoitustyö_vesanan (Tarkastettu)

Testiryhmä: Treeniähh (Tarkastettu)

Testiryhmä: Harjoitustyö_lehtau (Tarkastettu)

Testiryhmä: Harjoitustyö_juntume (Tarkastettu)

Testiryhmä: Harjoitustyö_alhoei (Tarkastettu)

KUVA 3. Etusivu

6.1 Hoitosuunnitelma

Alkuperäisessä järjestelmässä ollutta hoitosuunnitelma komponenttia haluttiin kehittää yksinkertaistamalla sen kirjaamista. Opiskelijan näkymä koki paljon muutoksia ja sen kehittämiseen varattiin paljon resursseja. Aluksi kirjataan potilaan perustiedot. Tähän näkymään ei tehty muutoksia. Seuraavaksi syötetään työn tarve-, tavoite- ja toimintotiedot. Ennen tiedot täytyi syöttää yksitellen, jokainen omasta näkymästään. Myös FinCC-luokituksen mukaiset polut tarve- ja toimintoluokituksille syötettiin yksi taso (komponentti, pääluokka, alaluokka) kerrallaan. Tätä kehitettiin niin, että kokonaisuus (tarve-, tavoite- ja toimintotiedot) pystytään antamaan samasta näkymästä ja tarvittaessa toimintotietoja pystytään lisäämään erillisestä näkymästä.

Vanhassa järjestelmässä kirjattiin vain yksi tavoitekenttä, joka oli kaikille tarpeille yhteinen. Päivityksen yhteydessä se muutettiin tarvekohtaiseksi eli jokaiselle tarpeelle on olemassa oma tavoitekenttä. Kun tarve-, tavoite- ja toiminto tiedot on syötetty järjestelmään, voidaan kirjata tehdyt toimenpiteet toteutus-kohtaan. Jokaiselle suunnitellulle toiminnolle voidaan kirjata oma toteutus (kuva 4).

Työtietojen muokkaus

Tarve

Komponentti:	Pääloukka:	Alaluokka:	Kirjoita kuvaus: (max. 1100 merkkiä) merkkejä jäljellä	Tulosluokitus:
5000: Aktiviteetti	5001: Aktiviteetin muutos	5002: Aktiviteetin heikkeneminen		HUO : Huonontunut

Tavoite (max. 300 merkkiä)

Tavoite potilaan näkökulmasta

Toiminnot

Komponentti:	Pääloukka:	Alaluokka:	Kirjoita kuvaus: (max. 1110 merkkiä)
1000: Aktiviteetti	1002: Aktiviteettia edistävä toiminta	1003: Leikki	

Tallenna ja lisää uusi toiminto (useampi toiminto) Valmis. Tallenna työhön: Kokeilutyö1 (vain yksi toiminto)

Peruuta

KUVA 4. Työn tietojen lisääminen (tarve, tavoite, toiminnot)

Työn tulostusnäkömään tehtiin myös suuria muutoksia. Vanha vaakatasoinen ja laatikkomallinen korvattiin pystysuuntaisella toteutuksella. Lisäksi sisältö jaoteltiin uudelleen. Tarve, tavoite, toiminto ja toteutus eriteltiin aiemmin omiksi kokonaisuuksiksi, mutta tämä vaihdettiin kokonaisuuden (tarve, tavoite, toiminto ja toteutus) kattavaan näkömään työn lukemisen helpottamiseksi. Tulostusnäkömässä työtä pystyy tarkastelemaan ilman tulostamista (kuva 5). Samassa näkömässä näkyy myös työstä annetut arviointit ja perustelut arvosanalle. Tulostetussa versiossa arviointeja ei näy.

Potilashoitosuunnitelman nimi: Testipotilas	
Hoitosuunnitelman tekijä:	18.09.2012, klo 08:50:21 0000000 - TESTI Opiskelija opiskelija@health.tamk.fi
Potilaan esitiedot:	
Nimi:	Veijo Virtanen
Sukupuoli:	mies
Ikä:	60 vuotta
Hoitoon hakeutumisen syy:	Verivirtsaisuus, jonka vuoksi hakeutunut hoitoon. Oikea munuainen poistettu syövän vuoksi. Nyt ensimmäinen päivä leikkauksesta.
Terveystottumukset ja -käyttäytyminen:	Perusterve
Fysiologiset mittaukset:	Paino 80 ,pituus 180, RR 135/80, p 100
Aikaisemmat sairaudet:	Virtsateinfektio 1980
Lääkehoito:	Perusterve, ei lääkityksiä
Potilaan terveyteen liittyvät riskitekijät:	Ei tiedossa
Potilashoitosuunnitelman nimi: Testipotilas	
Hoitosuunnitelman tekijä:	18.09.2012, klo 08:50:21 0000000 - TESTI Opiskelija opiskelija@health.tamk.fi
Työn tiedot:	
Hoidon tarve <ul style="list-style-type: none"> • Aktiiviteetti => Aktiiviteetin muutos => Liikkumisen rajoittuminen <i>Leikkauksesta johtuen liikkumisen rajoittuminen</i>	
Hoidon tavoite <i>Nousee ylös 1 pop avustuksella Kykenee sen jälkeen liikkumaan itsenäisesti</i>	
Suunnitellut toiminnot <ul style="list-style-type: none"> • Aktiiviteetti => Aktiiviteettiin liittyvä ohjaus <i>Oikea yläsnousutekniikka ja vuoteseen meno ohjataan</i>	
Toteutus <i>Nousi ylös ensimmäisenä leikkauksen jälkeisenä aamuna. Jalat liikkuvat hyvin. Tilanne (-ei luokitusta-)</i>	

KUVA 5. Työn tiedot

Työ voidaan palauttaa arvioitavaksi opettajan ohjeiden mukaiseen ryhmään. Palautuksen jälkeen työtä ei voi muokata uudessa järjestelmässä. Työ lukitaan siinä muodossa kun se palautus hetkellä on ja jätetään opettajalle odottamaan arviointia. Ainoastaan palautetut työt voidaan arvioida. Työ voidaan myös poistaa missä vaiheessa tahansa. Tällöin kaikki kyseisen työn tiedot ja mahdolliset arvioinnit poistetaan. Poisto vaatii käyttäjän vahvistuksen.

Opettajan näkymään tehtiin myös muutoksia. Työn arviointi kehitettiin sellaiseksi, että samassa näkymässä ovat työn tiedot, aiemmin mahdollisesti annetut arvosanat ja perustelut sekä arviointilomake arviointia varten (kuva 6). Opettaja pystyy antamaan arvosanan työlle, joka on palautettu mutta sitä ei ole arvioitu tai joka on saanut hylätty arvioinnin. Hylätty arviointi avaa työn opiskelijalle muokattavaksi. Hyväksytysti suoritettuja töitä ei voi enää uudelleen arvioida. Vanhassa järjestelmässä tallennettiin vain yksi arviointi. Uuteen versioon arviointien määrää kehitettiin niin, että arviointeja voi antaa rajattoman määrän.

Työn arviointi:	
Arvosana: hyväksytyt numerot 1-5 hyväksytyt suoritus = S hylätyt hylätty suoritus = H numero 0	-valitse-
Vapaamuotoinen perustelu: max: 4000 merkkiä (n. puoli A4 sivua)	
Arvioinnin tekijä:	(peljuk) Jukka Pelander
	Tallenna arviointi

KUVA 6. Työn arviointi

6.2 Yhteenveto

Uutena komponenttina järjestelmään toteutettiin yhteenveto. Yhteenveto komponentti oli prioriteetiltaan tärkeä toteuttaa kehitystyön aikana, joten tähän meni myös paljon resursseja. Yhteenveto laaditaan potilaasta hänen hoidon loppuvaiheessa tai kun potilas siirretään toiseen hoitoyksikköön. Yhteenvedon tarkoituksena on selvittää lyhyesti potilaan siirtyessä toiseen hoitoyksikköön, miten potilasta on hoidettu tähän saakka.

Opiskelija voi luodessaan uutta yhteenvetoa valita, aloittaako tyhjältä pohjalta vai otaanko tiedot jostain tehdystä hoitosuunnitelmasta. Hoitosuunnitelmasta tehdyssä yhteenvedossa haetaan hoitosuunnitelmasta kirjattujen tarpeiden ja toimintojen komponenttitason luokitukset. Yhteenvedon ensimmäisessä näkymässä kirjataan potilaan perustiedot ja hoidon arviointi kohdassa loppuarvioon kirjoitetaan lyhyesti arvio annetusta hoidosta. Hoidon tarve, -toiminnot ja -tulokset kohtiin kirjataan komponenttitason luokitus ja lyhyt kuvaus tiivistettynä. Lääkehoito kohtaan kirjataan lyhyesti annetut lääkkeet ennen toiseen hoitoyksikköön siirtymistä. Yhteenveto palautetaan tarkistettavaksi palauta painikkeella. Jos yhteenvetoa ei palauteta, se säilyy luonnoksena opiskelijalla. Poisto tapahtuu samalla tavalla kuin hoitosuunnitelman poisto eli käyttäjän täytyy vah-

vistaa poisto. Poistettaessa yhteenvedoa sen kaikki tiedot ja arvioinnit poistetaan järjestelmästä. Yhteenvedosta pyrittiin tekemään paperisen ohjeen kaltainen ja siinä onnistuttiin todella hyvin (kuva 7).

Yhteenvedo

Työ: Kokeilutyö1(yhteenvedo)	
Yhteenvedon tekijä:	22.08.2012, klo 08:23:59 0000000 - TESTI Opiskelija opiskelija@health.tamk.fi
Potilas	
Nimi ja ikä Matti Meikäläinen	
Hoidon arviointi	
Loppuarvio Tekstiä	
Hoidon tarve	Aktiviteetti Tekstiä
Hoitotyön toiminnot	Aktiviteetti Tekstiä
Hoidon tulokset	Aktiviteetti Tekstiä Tilanne Ennallaan
Lääkehoito Lääkkeet annettu klo 7.	
16.5.2013 TESTI Opiskelija Opiskelija	
Työstä saatu arviointi:	
Arvosana:	5
<u>hyväksytyt</u> numerot 1-5 hyväksytty = S	
<u>hylätyt</u> hylätty = H numero 0	
Perustelu:	Loistavaa työtä!
Arvioija:	(peljuk) Jukka Pelander
Arvioitu:	24.09.2012 09:30:52

KUVA 7. Yhteenvetokomponentti

Opettajanäkymässä työtä voi tarkastella tulostettavassa muodossa sekä arviointimuodossa. Tulostettavassa versiossa näkyy pelkkä yhteenvedo ja arvioitavassa versiossa näkymä on hyvin samantyylinen kuin hoitosuunnitelmaa arvioitaessa eli näkymässä on yhteenvedo, mahdollisesti annetut arvioinnit ja arviointilomake. Arviointeja pystyy yhteenvedoistakin antamaan rajattomasti ja vain palautettuja sekä hylättyjä yhteenvedoja voidaan arvioida.

6.3 Muita muutoksia

Tärkeimpien komponenttien muutostöiden lisäksi järjestelmään tehtiin myös muita muutoksia. Järjestelmässä oli toimintoja, joita voitiin muokata monesta eri paikasta. Pohdinnan jälkeen päädyttiin ratkaisuun, jossa samankaltaisten toimintojen hallinta niputettiin yhteen, samaan paikkaan, muokkaamisen helpottamiseksi ja järjestelmän selkeyttämiseksi. Järjestelmään jouduttiin kehittämään uusi tietokanta vanhan pohjalle, koska vanha versio ei vastannut tarpeeseen. Uusi tietokanta suunniteltiin erittäin käytännöllisen hyvällä MySQL Workbench –ohjelmalla (liite 1).

Uudistettuun järjestelmään toteutettiin määräaikaisten käyttäjien hallinta. Määräaikaiset käyttäjät syötetään järjestelmään samalla tavalla kuin muutkin käyttäjät. Käyttäjille valitaan käyttäjätaso ja lisätietona syötetään päivämäärä, johon saakka tunnus on käytettävissä. Tunnus voidaan uusia muokkaamalla päivämäärää.

Töiden ja arviointien selaus tapaa ja näiden tarpeellisuutta mietittiin pitkään ja päädyttiin ratkaisuun, jossa molemmat säilytetään. Selaus tapa haki pitkään muotoaan ja lopulta päädyttiin ratkaisuun, jossa selaus tapahtuu käyttäjäkohtaisesti eli opettaja voi selata valitun opiskelijan töitä tai arviointeja.

JavaScript valittiin tähän projektiin, koska sillä saatiin helposti lisättyä lomakkeisiin tarkastuksia ja vahvistuksia, esimerkiksi poistotoimintoihin. Järjestelmästä haluttiin helppokäyttöisempi jättämällä pois vaiheittainen luokitusten lisääminen. Tämä ohjasi valintaa JavaScriptiin. Myös se, että JavaScriptiä käytetään miljardeilla sivuilla, sai minut vakuuttuneeksi tekniikan hyödyllisyydestä. Käyttäjäystävällisyyden lisäämiseksi JavaScriptiä tarvittiin palvelimen kanssa kommunikointiin ilman, että sivuja ei tarvitse joka kerta kokonaan ladata uudestaan. Tämä tekniikka on nimeltään AJAX ja sen toteuttaminen vaatii JavaScriptin osaamista.

7 KÄYTTÖÖNOTTO JA OHJEISTUS

Järjestelmä otettiin käyttöön elokuun 2012 alussa. Yhdessä ohjaajien kanssa käytiin järjestelmän toiminta läpi, jotta sieltä ei löydy puutteita. Tehdyt muutokset hyväksyttiin ja kehitysympäristössä kehitetty järjestelmä siirrettiin tuotantokäyttöön. Suurimmaksi hidasteeksi osoittautui vaadittavien oikeuksien saanti siirtoa varten. Tiedostojen ja tietokannan siirto tapahtui ilman suurempia ongelmia ja järjestelmä oli elokuun puolessa välissä testikäytössä. Tässä vaiheessa järjestelmään avattiin sähköinen palaute, jonka kautta järjestelmän käyttäjien on mahdollista antaa palautetta sähköisesti nimellä tai nimettömästi järjestelmän kehittäjälle. Tätä kautta kerätään myös järjestelmän kehitysehdotuksia suoraan käyttäjiltä.

Käyttöönoton jälkeen järjestelmän käytöstä järjestettiin opettajille kolme ohjeistustilaisuutta, jotta mahdollistettiin mahdollisimman monen opettajan pääsy tilaisuuteen. Tilaisuuden aikana käytiin järjestelmä läpi opiskelijan- ja opettajan näkymästä ohjeistuksen painottuessa opettajan näkymään, erityisesti töiden arviointiin. Saadun palautteen perusteella tilaisuus järjestettiin onnistuneesti. Töiden arviointitapaa pidettiin myös onnistuneena uudistuksena.

Syksyllä 2012 aloittaneille opiskelijoille järjestettiin myös järjestelmän ohjeistustilaisuuksia. Tilaisuudet järjestettiin yhdessä ryhmän ohjaajan kanssa ja tilaisuudessa käytiin yksityiskohtaisesti läpi opiskelijan näkymästä hoitosuunnitelmien tekeminen. Tavoitteena oli, että opiskelijat pystyvät tämän jälkeen itsenäisesti tekemään hoitosuunnitelman annetun case harjoituksen pohjalta tai itse hoitamastaan potilaasta. Tilaisuudet onnistuivat hyvin ja tavoitteeseen päästiin, sillä opiskelijat osasivat syöttää tiedot järjestelmään ja tulostaa työn.

8 YHTEENVETO JA JATKOKEHITYS

Suuremmilta ohjelmointivirheiltä vältyttiin ja järjestelmä saatiin kehitettyä valmiiksi ajoissa ennen julkaisuajankohtaa. Tuotteen elinkaareen kuuluu myös tuotteen jatkokehittäminen käyttäjiltä saadun palautteen perusteella. Järjestelmästä kerättiin sähköisesti palautetta opiskelijoilta ja opettajilta. Palautetta pystyi antamaan nimellä tai nimettömänä. Kehitysehdotuksia ja palautteita saatiin 26 opiskelijalta ja yhdeltä opettajalta sähköisen palautelomakkeen kautta (liite 2).

Palautteita analysoitiin ensin sen perusteella, että oliko palaute annettu nimellä vai nimettömänä. Yleisesti arvioituna nimettömänä annettu palaute sisälsi kritiikkiä ilman perusteita ja oli jossain määrin asiatonta, kun taas nimellä annettu palaute oli rakentavaa ja sisälsi hyviä kehitysehdotuksia.

Järjestelmän tekniikka uusittiin melko perin pohjin ensimmäisessä kehitysvaiheessa mutta edelleen jatkokehitysajatuksina ajateltiin, että ulkoasun parannus/kehitys on graafikon työtä ja siihen työhön on syytä valita ulkoasusuunnitteluun erikoistunut henkilö. Tietoteknisen henkilön vastuulle ajateltiin kirjata tekniikan kehittäminen saadun palautteen perusteella.

Saadun palautteen perusteella työn kirjaamisvaiheetta tulee kehittää, jotta kaikkia kohtia ei tarvitsisi kirjata samalla kertaa ennen tallennusta. Tärkeä päivitys on myös hoidon tarpeen poistonapin logiikan teko. Tietokantaan tallennettavien tietojen merkkimäärät todettiin myös tarkastettaviksi ja tarvittaessa muutettava suuremmiksi. Osa sijoitteluista ei toimi oikein Internet Explorer –selaimella, koska IE tekee toiset asiat eri tavalla kuin Mozillan Firefox ja Googlen Chrome –selaimet. Tämän vuoksi ohjelmaan on asetettu selainsuosituksiksi Firefox ja Chrome. Myös IE:n selaimen saamista samaan listaan tulee tutkia eli osa css-tyylitiedoston arvoista täytyy määrittää IE selaimelle erikseen. Reaaliaikaisen tallennuksen mahdollisuutta haluttiin myös tutkittavan, mikäli aikaa ja resursseja on riittävästi.

LÄHTEET

CWE/SANS Top 25 Software Errors. 2011. Luettu 28.2.2013

<http://cwe.mitre.org/top25/#CWE-89>

Haikala, I.;& Mikkonen, T. 2011. Ohjelmistotuotannon käytännöt (12. painos).

Talentum.

Immonen, J. 2003. Luentomoniste. Luettu 27.2.2013

https://cs.joensuu.fi/~jimmonen/jot_moniste/pics/waterfall.gif

Notepad++ Home. 2011. Luettu 7.3.2013

<http://notepad-plus-plus.org/>

MySQL. 2013. Luettu 28.2.2013

<https://www.mysql.com/about/>

MySQL Workbench 5.2. 2012. Luettu 7.3.2013

<https://www.mysql.com/products/workbench/>

PHP. 2013. Luettu 28.2.2013

<http://php.net/>

putty.org. 2008. Luettu 8.3.2013

<http://www.putty.org/>

Wikipedia, ketterä ohjelmistokehitys. 2013. Luettu 27.2.2013

https://fi.wikipedia.org/w/index.php?title=Ketter%C3%A4_ohjelmistokehitys&oldid=12555669

Wikipedia, PHP. 2013. Luettu 28.2.2013

<https://fi.wikipedia.org/w/index.php?title=PHP&oldid=12701340>

Wikipedia, suihkulähdemalli. 2012. Luettu 27.2.2013

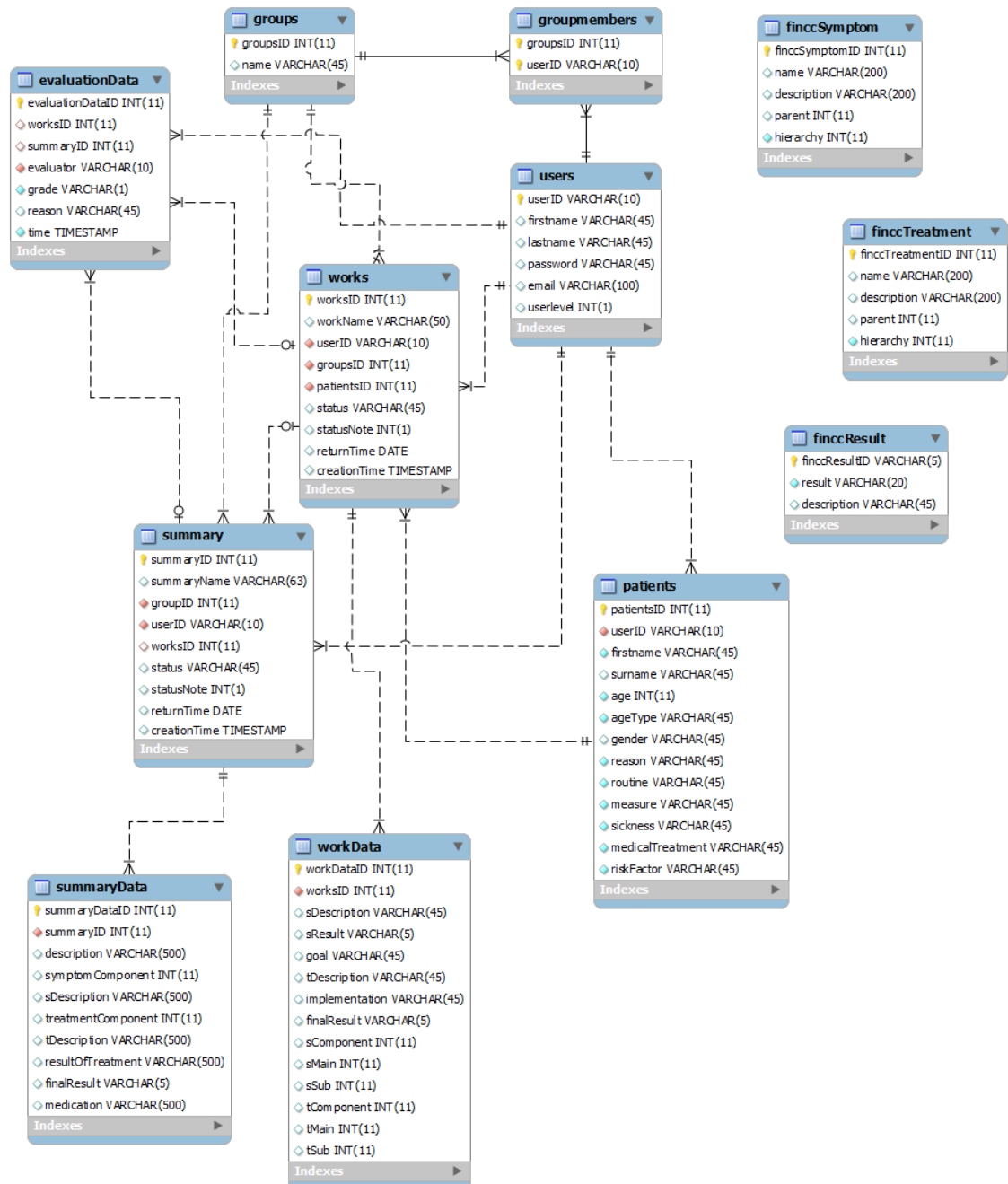
<https://fi.wikipedia.org/w/index.php?title=Suihku%C3%A4hdemalli&oldid=12251866>

W3School - JavaScript. 2010. Luettu 7.3.2013

<http://www.w3schools.com/js/default.asp>

LIITTEET

Liite 1. Järjestelmän ER-kaavio



Liite 2. Tilasto Nurseuksesta

Tilastoja

Tilasto elokuu 2012 => toukokuu 2013

Tehtyjen hoitosuunnitelmien määrä yhteensä:	1463 kpl
- hoitosuunnitelmia järjestelmässä nyt:	1357 kpl
- palauttamattomia(luonnokset):	654 kpl
Tehtyjen yhteenvetojen määrä yhteensä:	205 kpl
- yhteenvetoja järjestelmässä nyt:	201 kpl
- palauttamattomia(luonnokset):	148 kpl
Arviointien määrä:	547 kpl
- hylättyjä:	165 kpl
Käyttäjiä järjestelmässä:	1134 kpl
- oppilaita:	1036 kpl
- opettajia:	91 kpl
- pääkäyttäjiä:	7 kpl
Palautteita annettu:	27 kpl