



# **NURSEUS-JÄRJESTELMÄN JAT- KOKEHITYS**

Tero Sarttila

Opinnäytetyö  
Toukokuu 2014  
Tietotekniikan koulutusohjelma  
Ohjelmistotekniikka

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tietotekniikka  
Ohjelmistotekniikka

TERO SARTTILA:

Nurseus -järjestelmän jatkokehitys

Opinnäytetyö 31 sivua, joista liitteitä 2 sivua  
Toukokuu 2014

---

Nurseus-järjestelmä on kehitetty Tampereen Ammattikorkeakoulun sosiaali- ja terveystieteiden opiskelijoille. Nurseus-järjestelmä toimii oppimisympäristönä, jossa harjoitellaan hoitotyön dokumenttien tekoa rakenteisen mallin mukaan. Tässä dokumentissa käsitellään tietojärjestelmien kehitystä yleisesti sekä kuvataan opinnäytetyönä toteutettua Nurseus-järjestelmän jatkokehitystä ja tekniikoita, joilla se toteutettiin.

Nurseus -järjestelmän jatkokehityksen aikana järjestelmästä tehtiin käyttäjäystävällisempi useiden muutosten avulla. Järjestelmän näkymä jaettiin välilehtiin asiakokonaisuuksien perusteella ja suurin osa järjestelmän toiminnasta muutettiin dynaamiseksi AJAX-tekniikan avulla. Hoitosuunnitelmien FinCC-luokituskokonaisuuksien määrittämistä helpotettiin hakukoneen avulla. Uutena komponenttina järjestelmään lisättiin lomake, jossa voidaan harjoitella lääkehoidon tietojen kirjaamista.

Nurseus -järjestelmän jatkokehitys oli onnistunut ja sille asetettuihin tavoitteisiin päästiin, mutta jatkokehitykselle on edelleen tarvetta, koska oppimisympäristönä Nurseus-järjestelmän kaikkia mahdollisuuksia ei ole vielä löydetty. Lisäominaisuuksien tarpeita ilmenee jatkuvasti käyttäjien antamien palautteiden muodossa.

## **ABSTRACT**

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

Tero Sarttila:  
Further development of the Nurseus system

Bachelor's thesis 31 pages, appendices 2 pages  
May 2014

---

Nurseus system has been developed for Health and Social Care students at Tampere University of Applied Sciences. Nurseus system is learning environment for making nursing documents. This document elaborates development of information system in general as well as describes further development of Nurseus system and techniques used for implementation.

Several changes in further development made the system more user-friendly. View of the system was divided into tabs on the basis of content and most of the content was changed to dynamic with AJAX. Determining FinCC-ratings were made easier by implemented search engine. As a new component, form was made for students to practice their medical documentation skills.

The further development of Nurseus was successful and its objectives were achieved but further development is still needed because Nurseus full potential is not found yet. The need for additional features is constantly discovered from user's feedbacks.

---

Key words: nurseus, information system, php, jquery, ajax

## SISÄLLYS

1	JOHDANTO.....	6
2	NURSEUS-JÄRJESTELMÄ .....	7
3	TIETOJÄRJESTELMÄN KEHITTÄMISEN VAIHEET .....	9
3.1	Esitutkimus .....	9
3.2	Vaatimusmäärittely .....	10
3.3	Suunnittelu .....	10
3.4	Toteutus .....	11
3.5	Testaus .....	11
3.6	Tietojärjestelmän käyttöönotto .....	12
4	KÄYTETYT TEKNIIKAT JA TYÖKALUT .....	14
4.1	HTML-merkintäkieli .....	14
4.2	CSS-tyylimäärittely .....	15
4.3	PHP-kieli.....	16
4.4	JavaScript-kieli .....	17
4.4.1	JQuery-kirjasto.....	17
4.4.2	AJAX-tekniikka .....	18
4.5	SQL-kyselykieli ja MySQL-tietokanta.....	19
4.6	Notepad++-tekstieditori .....	19
4.7	Firebug-liitännäinen.....	20
4.8	MySQL Workbench -ohjelma.....	21
5	PÄIVITETTY NURSEUS-JÄRJESTELMÄ.....	22
5.1	Nurseus-järjestelmän arkkitehtuuri.....	22
5.2	Välilehtinäköymä.....	22
5.3	Hakutoiminto .....	24
5.4	Lääkehoito .....	25
5.5	Yhteenvedon muutokset .....	25
5.6	Aikakatkaistu .....	27
5.7	Käyttö- ja asennusohjeet.....	27
6	YHTEENVETO JA NURSEUS-JÄRJESTELMÄN TULEVAISUUS .....	28
	LÄHTEET.....	29
	LIITTEET .....	30
	Liite 1. Aiheeseen liittyviä linkkejä. ....	30
	Liite 2. Nurseus -järjestelmän ER-kaavio .....	31

**LYHENTEET JA TERMIT**

AJAX	Tekniikka, joka käyttää JavaScript- ja XML -tekniikoita lähettämään ja palauttamaan dataa selaimen ja web-palvelimen välillä asynkronisesti (Asynchronous JavaScript and XML)
CSS	HTML-sivun tyylimäärittely (Cascading Style Sheets)
ER-kaavio	Kuvaus tietokantojen välisistä yhteyksistä (Entity-relationship modeling)
FinCC	Hoitotyön kansallinen luokitusjärjestelmä (Finnish Care Classification)
HTML	Verkkosivun merkintäkieli (Hypertext Markup Language)
jQuery	JavaScriptin lisäkirjasto
MVC	Ohjelmistoarkkitehtuurityyli, jonka tarkoituksena on käyttöliittymän erottaminen sovellusalueiedoista (Model-View-Controller)
MySQL	Avoimen lähdekoodin relaatiotietokanta
Parametri	Ohjelmalle käynnistyksen yhteydessä välitettävä tieto ja ohjelmoinnissa funktiolle tai käskylle välitettävä tieto
SQL	Standardoitu kyselykieli relaatiotietokannoille (Structured Query Language)
Syntaksi	Lauseoppi
URL-osoite	Web-sivun, kuvan tai äänitiedoston kaltaisen resurssin sijainti (Uniform Resource Locator)
Web-sivu	Web-sivu eli verkkosivu tarkoittaa internetissä julkaistua sivua
Web-palvelin	Tietokone tai ohjelmisto, joka jakaa dokumentteja HTTP-protokollalla asiakasohjelmille ja koneille

## 1 JOHDANTO

Terveydenhuollon dokumentit siirtyvät kansalliseen eArkistoon syyskuusta 2014 alkaen. Sähköinen arkistointimuoto vaatii kaikille yhteiset rakenteet, jotka mahdollistetaan kaikissa käytössä olevissa potilastietojärjestelmissä. eArkiston rakenteina käytettävistä koodistoista vastaa Terveyden- ja hyvinvoinnin laitos. (Nurseusinfo 2014)

Hoitotyötä opettavien ammattikorkeakoulujen ja ammatillisten oppilaitosten haasteena on tuottaa hoitotyön opiskelijalle kansallisen rakenteen vaatimat valmiudet laatia hoitotyön dokumentteja. Nurseus on selainpohjainen kirjaamisalusta, jossa voi turvallisesti suljetussa oppimisympäristössä harjoitella hoitotyössä tarvittavia, dokumentointiin liittyviä taitoja. (Nurseusinfo 2014)

Tämä opinnäytetyö keskittyy Nurseus-järjestelmän jatkokehitykseen. Tärkeimpinä tavoitteina oli saada järjestelmästä käyttäjäystävällisempi ja korjata tiedossa olevia vikoja järjestelmästä.

Opinnäytetyön luvussa kaksi käsitellään Nurseus-järjestelmän toimintaa. Luvussa kolme käydään läpi tietojärjestelmien kehittäminen yleisesti. Luvussa neljä käydään läpi järjestelmän kehityksessä käytettyjä tekniikoita sekä työkaluja, joilla toteutukset tehtiin. Luvussa viisi kerrotaan jatkokehityksessä tehdyistä parannuksista järjestelmään. Lisäksi liitteeseen 1 on kerätty linkkejä materiaaleihin, jotka liittyvät tähän opinnäytetyöhön. Näitä materiaaleja ei ole käytetty suoranaisesti lähteinä tämän dokumentin tekemiseen.

## 2 NURSEUS-JÄRJESTELMÄ

TAMK:n sosiaali- ja terveystieteiden opiskelijoilla on käytössä web-pohjainen Nurseus-järjestelmä, jolla voidaan harjoitella hoitotyön dokumentointiin liittyviä taitoja, kuten hoitosuunnitelman ja hoitosuunnitelman yhteenvedon tekoa. Web-pohjaisen järjestelmän hyötynä on, että harjoitustöiden tekeminen ei ole sidoksissa vain oppilaitoksessa tehtäviin harjoituksiin vaan opiskelijat pystyvät harjoittelemaan dokumentointia niin kotona kuin työharjoittelussa.

Harjoitustyöt voivat liittyä hoitotyön harjoittelutunteihin, case-pohjaiseen dokumentoinnin harjoitteluun tai harjoittelujaksojen hoitotyön suunnitelmien dokumentointiin. Nurseus toimii oppimisympäristönä kansallisen mallin mukaiseen dokumentointiin ja antaa opiskelijoille ensikosketuksen työelämässä käytettäviin potilastietojärjestelmiin.

Nurseus-järjestelmän hoitosuunnitelmaan kirjataan potilaan esitiedot, hoidon suunnittelu sekä hoidon tavoite ja toiminnot. Esitietoihin kuuluu potilaan yksilöintitiedot, hoitoon hakeutumisen syyt, terveystottumukset, aikaisemmat sairaudet, lääkitykset ja terveyteen liittyvät riskitekijät kuten allergiat. Nurseus-järjestelmän suunnittelussa on otettu huomioon rakenteisen kirjaamisen malli ja hoitosuunnitelman perustana on FinCC-luokitusjärjestelmä.

FinCC-luokitus muodostuu suomalaisesta hoidon tarveluokituksesta (SHTaL), suomalaisesta hoitotyön toimintoluokituksesta (SHToL) ja suomalaisesta hoidon tulosluokituksesta (SHTuL). SHTuL sisältää taulukon 1 mukaiset luokitukset. SHTaL:lla ja SHToL:lla on yhtenevä hierarkkinen rakenne (komponentti-, pääluokka- ja alaluokkataso). (Liljamo, Kinnunen & Ensio, 2012, 3.)

TAULUKKO 1. SHTuL

Tunniste	Lyhenne	Määritelmä
EN	Ennallaan	Tilassa ei muutosta - ennallaan
HUO	Huonontunut	Tilanne muuttunut - huonontunut
PA	Parantunut	Tilanne muuttunut - parantunut

Esimerkiksi, jos potilaalla on kuumetta hoitoon tullessa, niin hoidon suunnittelun tarveluokituksiksi kirjattaisiin taulukon 2 mukainen tarveluokitus. Luokitusta täsmennetään hoitosuunnitelmaan vapaalla tekstillä. Esimerkiksi ”Kuumetta ollut kotona kolmena päivänä yli 39 astetta.” Mahdollinen toimintoluokitus tässä tapauksessa olisi taulukon 3 mukainen ja luokitusta voitaisiin täsmentää seuraavalla tavalla: ”Mittaus kaksi kertaa vuorokaudessa.”

TAULUKKO 2. SHTaL

<b>Komponentti</b>	<b>Pääluokka</b>	<b>Alaluokka</b>
Verenkierto	Kehon lämpötilan muutos	Kuume

TAULUKKO 3. SHToL

<b>Komponentti</b>	<b>Pääluokka</b>	<b>Alaluokka</b>
Verenkierto	Kehon lämpötasapainosta huolehtiminen	Lämmön seuranta

Nurseus-järjestelmässä hoitotyön yhteenvetoon kirjataan hoitojakson keskeiset ydintiedot ja lääketiedot. Yhteenveto on mahdollista tulostaa ja antaa potilaalle hoitojakson lopussa. Ydintiedoista selviää loppuarvio, hoidon tarve, hoitotyön toiminnot ja hoidon tulokset.

Nurseus-järjestelmässä opettaja pystyy laatimaan opiskelijoille malleja hoitosuunnitelmista, seuraamaan opiskelijoiden kehitystä heidän tekemistä harjoitustöistä, arvostelemaan harjoitustöitä sekä antamaan palautetta harjoitustöiden pohjalta.



### 3 TIETOJÄRJESTELMÄN KEHITTÄMISEN VAIHEET

Tietojärjestelmän kehittäminen alkaa tarpeesta kehittää uutta tai ylläpitää vanhaa. Usein kehitystyö käynnistyy asiakkaan tarpeen takia. Muita syitä voivat olla esimerkiksi uusien teknisten mahdollisuuksien hyödyntäminen tai jonkin toisen kehityshankkeen yhteydessä esiin tulleet tarpeet. (Pohjonen 2002, 26.)

Varsinainen tietojärjestelmien kehitystyö sisältää seuraavat tai osan seuraavista asioista:

- esitutkimus
- vaatimusmäärittely
- suunnittelu
- toteutus
- testaus
- tietojärjestelmän käyttöönotto

#### 3.1 Esitutkimus

Kun idea tietojärjestelmän kehittämisestä on syntynyt, on ensimmäisenä selvitettävä pystytäänkö hanke toteuttamaan. Tästä vaiheesta käytetään nimitystä esitutkimus.

Esitutkimuksessa ei vielä suunnitella teknisiä ratkaisuja taikka tehdä varsinaista ohjelmointityötä. Esitutkimuksen tehtävänä on tuottaa tietoa tietojärjestelmän kehittämisestä päättävälle sekä määrittää lähtökohdat mahdolliselle tietojärjestelmän kehityshankkeelle. Pohjosen (2002, 27) mukaan esitutkimusraportista selviää seuraavat asiat:

- organisaation tietojenkäsittelyn nykytilanne
- ongelmien kuvaukset, johon järjestelmän oletetaan tuottavan ratkaisut
- sidosryhmien kuvaukset, joita hanke koskee
- järjestelmälle asetettavien tavoitteiden alustavat määritykset
- uuden järjestelmän kehittämistavoitteiden määritykset
- eri toimintavaihtoehtojen kuvaukset arvioineen ja perusteluineen
- alustava suunnitelma tietojärjestelmän kehittämishankkeen läpiviemiseksi

Esitutkimuksen perusteella tehdään päätös järjestelmän kehittämisestä tai kehittämättä jättämisestä. Esitutkimus ei siis aina automaattisesti johda tietojärjestelmähankkeen käynnistämiseen, vaan esitutkimuksen perusteella voidaan myös hylätä kaavailtu hanke. (Pohjonen 2002, 27.)

### **3.2 Vaatimusmäärittely**

Vaatimusmäärittelyksi kutsutaan dokumenttia, joka kuvaa järjestelmälle eri sidosryhmien asettamat vaatimukset. Vaatimusmäärittelyssä kuvataan eri sidosryhmien tarpeet järjestelmän suhteen, mutta ei oteta kantaa tekniseen toteutukseen. Järjestelmän toiminnallisia vaatimuksia kuvataan yleensä käyttötapauskaavioiden avulla. Vaatimusmäärittely luodaan asiakkaan kanssa yhteistyössä. (Pohjonen 2002, 28.)

Nurseus-järjestelmän jatkokehityksen vaatimusmäärittely laadittiin asiakkaan kanssa yhteistyössä käyttäjiltä saadun palautteen perusteella. Vaatimusmäärittelyä tarkennettiin asiakkaan pyynnöstä projektin edetessä moneen kertaan, koska käyttäjiltä saatiin aktiivisesti palautetta jatkokehityksen aikana. Tämä aiheutti ylimääräistä työtä, koska joitakin asiakokonaisuuksia jouduttiin toteuttamaan kahteen kertaan. Jos vaatimusmäärittelyyn olisi käytetty enemmän resursseja, projektin kokonaisaika olisi lyhentynyt. Tämän toteutustavan johdosta päästiin kuitenkin lähemmäksi loppukäyttäjän haluamaa lopputulosta, koska loppukäyttäjien palaute vaikutti muutoksiin jatkuvasti.

### **3.3 Suunnittelu**

Kun esitutkimuksessa ja vaatimusmäärittelyssä on saatu kuvattua, miksi järjestelmä rakennetaan ja mitä järjestelmä tekee, on seuraavana vaiheena suunnitella miten järjestelmä toteutetaan. Suunnittelun tarkoituksena on muuntaa vaatimusmäärittelyn aikana tehty toiminnallinen määrittely tekniseksi määrittelyksi, joka kuvaa järjestelmän toteutuksen. (Pohjonen 2002, 32.)

Tekninen määrittely kuvaa seikkaperäisesti järjestelmän toteutuksen ja arkkitehtuurin. Siitä selviää esimerkiksi millä ohjelmointikielillä eri moduulit rakennetaan sekä niiden rakenne ja keskinäinen hierarkia.

Nurseus-järjestelmän jatkokehityksen aikana suunnitteluun käytettiin resursseja aika vähäinen määrä, koska isoimmat tekniset ratkaisut oli jo tehty alkuperäisen järjestelmän toteutuksen yhteydessä, kuten ohjelmointikielien valinta. Näitä valintoja ei lähdetty muuttamaan, mutta joitakin lisäkirjastoja tuotiin helpottamaan järjestelmän toteutusta, kuten JavaScriptin kirjasto jQuery. jQuery-kirjastosta kerrotaan tarkemmin käytetyt tekniikat osiossa ja liitteeseen 1 on kerätty linkkejä jQuery-aiheisiin sivustoihin.

### 3.4 Toteutus

Suunnittelua seuraa toteutus. Toteutusvaiheessa ohjelmisto tai sen osa toteutetaan ohjelmointikielillä tai sovelluskehittimellä. Lopuksi ohjelmamoduulit integroidaan toimivaksi kokonaisuudeksi. Mikäli aiemmat vaiheet on suoritettu asianmukaisesti, pitäisi toteutuksen olla melko suoraviivainen toimenpide: kaikki tärkeimmät ohjelmiston rakennetta ja toiminnallisuutta koskevat ratkaisut on tehty jo aiemmin. (Pohjonen, 2002.)

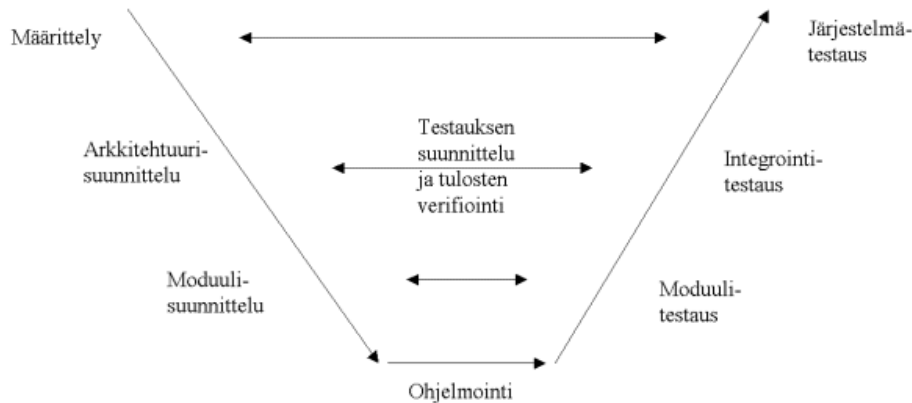
Nurseus-järjestelmän jatkokehityksen toteutusta varten luotiin oma kehitysympäristö, joka oli kopio alkuperäisestä järjestelmästä ja sisälsi testitietokannan. Muutokset toteutettiin ensin kehitysympäristössä ja käyttötapaustestauksen jälkeen ne siirrettiin asiakkaan järjestelmään. Käyttöönotto tapahtui vaiheittain: aina kun yksi asiakokonaisuus oli toteutettu ja testattu, niin se otettiin myös käyttöön asiakkaan järjestelmässä. Toteutusta tehtiin ketterän menetelmän mallin mukaisesti. Liitteessä 1 on linkkejä ketterästä menetelmästä kertoviin materiaaleihin.

Itse ohjelmointityö tehtiin Notepad++-tekstieditorilla, joka sisältää syntaksiväriytyksen kaikille toteutuksessa käytetyille ohjelmointikielille. Vianetsinnässä hyödynnettiin FireBug-liitännäistä ja tietokannan muutokset sekä hallinnointi tehtiin MySQL Workbench -ohjelmalla. Näistä ohjelmista kerrotaan tarkemmin käytetyt tekniikat ja työkalut -osiossa sivulta 19 alkaen.

### 3.5 Testaus

Toteutettu ohjelmisto täytyy testata ennen kuin se voidaan ottaa käyttöön. Testauksen tarkoituksena on löytää ohjelmistosta virheitä sekä varmistaa, että järjestelmä toimii vaatimusmäärittelyn mukaisesti. Testaus tapahtuu yleensä monella tasolla ns. V-mallin mukaisesti (KUVA 1). V-mallissa testaus jaetaan moduulitestaukseen, integrointitesta-

ukseen ja järjestelmätestaukseen. Moduulitestauksessa etsitään vikoja yksittäisistä moduuleista, integrointitestauksessa moduulien yhteistoiminnasta ja järjestelmätestauksessa koko järjestelmän toiminnoista ja suorituskyvystä. V-mallin mukaisessa testauksessa järjestelmätestaus suunnitellaan osana ohjelmiston määrittelyä ja testaus tehdään vertaamalla valmista järjestelmää sen toiminnalliseen määrittelyyn. (Pohjonen 2002, 35-36.)



KUVA 1. V-malli (Haikala, 2011)

Nurseus-järjestelmän jatkokehityksen aikana testaus suoritettiin erittäin vähäisillä resursseilla toteutuksen yhteydessä. Testauksen aikana pyrittiin pääasiassa katsomaan vain, että toteutus vastaa asiakkaan määrittelemää tarvetta. Mahdollisten vikojen löytäminen jäi loppukäyttäjän harteille käyttöönoton jälkeen ja näitä vikoja pyrittiin korjaamaan mahdollisimman nopeasti annetun palautteen perusteella.

### 3.6 Tietojärjestelmän käyttöönotto

Kun järjestelmä on testattu, voidaan se ottaa käyttöön. Tietojärjestelmän käyttöönottoon liittyy tekijöitä, jotka tulee huomioida ja valmistella huolellisesti. Tällaisia seikkoja ovat mm. tietojen, tiedostojen ja tietokantojen siirtäminen uuteen järjestelmään. Myös aikaisempien tai rinnakkaisten järjestelmien olemassaolo on huomioitava. Keskeinen tehtävä on myös käyttäjien ja ylläpitohenkilökunnan kouluttaminen. Vähimmäisvaatimuksena voidaan pitää asianmukaista käyttöohjeistusta. (Pohjonen 2002, 37.)

Ennen Nurseus-järjestelmän käyttöönottoa järjestettiin hyväksymistestaus asiakkaan kanssa, jossa asiakas tarkasti että Nurseus-järjestelmä on asetettujen vaatimusten mukainen ja valmis käyttöönotettavaksi.

Nurseus-järjestelmän käyttöönotto tapahtui siirtämällä tiedostot asiakkaan web-palvelimelle sekä siirtämällä uusi tietokanta asiakkaan MySQL-palvelimelle MySQL Workbench -ohjelmaa käyttäen. Vanha Nurseus säilytettiin uuden rinnalla eri kansiossa, koska vanhan järjestelmän harjoitustyöt eivät olleet yhteensopivia uuden tietokannan kanssa ja opiskelijoiden arvioidut työt on säilytettävä tallessa. Uusi Nurseus otettiin käyttöön syyskuussa 2013 URL-osoitteessa: <http://nurseus.tamk.fi> ja vanha Nurseus toimii URL-osoitteessa: <http://nurseus2012.tamk.fi>. URL-osoitteita varten vaadittavat asetustiedostojen muutokset teki asiakkaan it-tuki, koska tämän opinnäytetyön tekijällä ei ollut vaadittavia oikeuksia palvelimelle. Käyttöönotto tapahtui ilman suurempia ongelmia ja käyttäjien palautteena ilmenneet pienet virheet korjattiin nopeasti käyttöönoton jälkeen.

Käyttöönoton yhteydessä ei järjestetty erillistä käyttökoulutusta, koska vanhan Nurseus-järjestelmän käyttöönoton yhteydessä opettajille oli järjestetty kolme koulutustilaisuutta ja uuden Nurseus-järjestelmän opetteluun vanhan pohjalta pitäisi riittää pelkät käyttöohjeet. Uudet käyttöohjeet laadittiin jatkokehityksen pohjalta ja ne sisällytettiin järjestelmään. Jatkokehityksen aikana TAMK:n lehtorit Helena Vesaluoma ja Taru Lehtimäki pääsivät myös opettelemaan uuden järjestelmän käyttöä asiakastapaamisissa ja pystyvät näin ollen jakamaan tietoa kollegoilleen tarvittaessa. Opiskelijoille uuden järjestelmän käyttö opastetaan opettajien toimesta harjoitusluennoilla.

## 4 KÄYTETYT TEKNIIKAT JA TYÖKALUT

Tässä luvussa kuvataan Nurseus-järjestelmän jatkokehityksessä käytettyjä tekniikoita ja työkaluja sekä esitetään yksinkertaisia esimerkkejä tekniikoiden käytöstä. Jatkokehityksen tekniikat perustuivat suurelta osin alkuperäisen järjestelmän tekniikoihin sekä aikaisempiin kokemuksiin kyseisistä tekniikoista, joten niitä ei lähdetty vaihtamaan.

Työkalut oli valittu ilmaiseen lähdekoodiin perustuvien ohjelmien joukosta, koska projektille ei ollut varattu resursseja maksullisiin ohjelmiin. Työkalujen valintakriteereinä olivat aikaisempien projektien tuoma kokemus sekä mieltymys pienikokoisiin sekä hyvin suoritustehoisiin ohjelmiin sen sijaan, että niissä olisi paljon ylimääräisiä ominaisuuksia heikentämässä käytettävyyttä ja suoritustehoa.

### 4.1 HTML-merkintäkieli

HTML on lyhenne sanoista Hypertext Markup Language eli hypertekstin merkintäkieli. HTML:llä voidaan merkitä hypertekstin rakenne eli esimerkiksi mikä osa tekstistä on otsikkoa ja mikä leipätekstiä. Käytännössä HTML on siis kaikkien web-sivujen perusta. (Matthew 2012, 12.)

Web-sivujen rakennetta kuvataan tagien avulla, joista on esimerkkejä taulukossa 4. Kuvassa 2 nähdään esimerkki web-sivun lähdekoodista ja malli mitenkä tageja käytetään. Kuvan 2 oikeassa laidassa nähdään miltä sivu näyttää selaimessa.

TAULUKKO 4. Esimerkkejä HTML-tageista ja niiden merkityksestä

Tagi	Merkitys
html	Dokumentin HTML-osa
head	Dokumentin otsikko-osa
body	Dokumentin runko
h1,h2,h3,h4,h5,h6	Tasojen 1-6 otsikot
p	Kappale
form	Lomake
input	Syöte
textarea	Tekstialue
href	Kohde johon linkki vie
table	Taulukko

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML Esimerkki</title>
5     <meta charset="utf-8">
6   </head>
7   <body>
8
9     <h1>Tämä on otsikko</h1>
10
11     <p>Tämä on kappale, joka sisältää tekstiä.</p>
12
13   </body>
14 </html>

```

## Tämä on otsikko

Tämä on kappale, joka sisältää tekstiä.

KUVA 2. Esimerkki HTML:n käytöstä

## 4.2 CSS-tyylimäärittely

Cascading Style Sheets eli CSS on yksinkertainen kieli, jolla voidaan määrittellä tyyli HTML-dokumentin elementeille eli CSS-tyylien avulla määritellään web-sivun ulkoasu, kun HTML:n avulla määritellään sen rakenne. CSS-tyylejä voidaan sisällyttää suoraan HTML-dokumenttiin tai ne voidaan koota yhtenäiseksi säännöstöksi erilliseen tyyli-tiedostoon web-sivustoa varten. Erillinen tyyli-tiedosto helpottaa sivuston ylläpidettävyyttä, koska kaikki ulkoasuun vaikuttavat tyyli löytyvät helposti samasta paikasta.

(Lunn 2012, 6.)

Kuvassa 3 on edelliseen esimerkkiin (KUVA 2) lisätty toinen kappale riveille 16–17 ja sitä on muotoiltu tyyli-tiedoston avulla, joka näkyy kuvan 3 oikeassa yläreunassa. Tämä kappale on rajattu muusta HTML-dokumentista div-tagin avulla. Tämän käyttö näkyy kuvassa riveillä 15 ja 18.

```

esimerkki.html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML Esimerkki</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" type="text/css" href="tyylit.css">
7   </head>
8   <body>
9
10    <h1>Tämä on otsikko</h1>
11
12    <p>Tämä on kappale, joka sisältää tekstiä.</p>
13
14
15    <div id="muotoilu">
16      <p>Tämä on toinen kappale, joka on muotoiltu
17        tyyli-tiedoston avulla.</p>
18    </div>
19
20  </body>
21 </html>

```

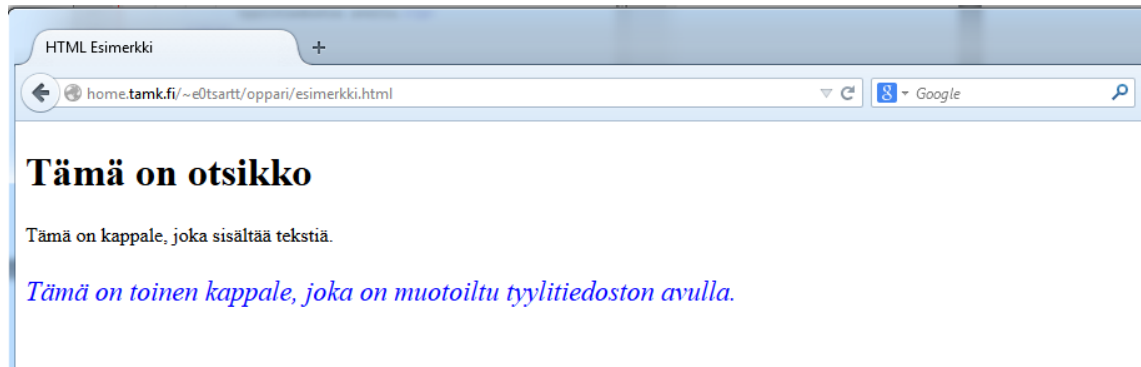
```

tyylit.css
1 #muotoilu{
2   font-size: 22px;
3   font-style: italic;
4   color: blue;
5
6 }
7

```

KUVA 3. Esimerkki HTML-dokumentin muotoilusta CSS-tyyli-tiedoston avulla

Kuvassa 4 nähdään miten kuvan 3 tyylitiedosto on muotoillut toisen tekstikappaleen fontin koon 22 kuvapisteen kokoiseksi, siniseksi ja muuttanut tämän tyylin kurssiiviksi.



KUVA 4. Tyylitiedoston tulokset selainäkymässä

### 4.3 PHP-kieli

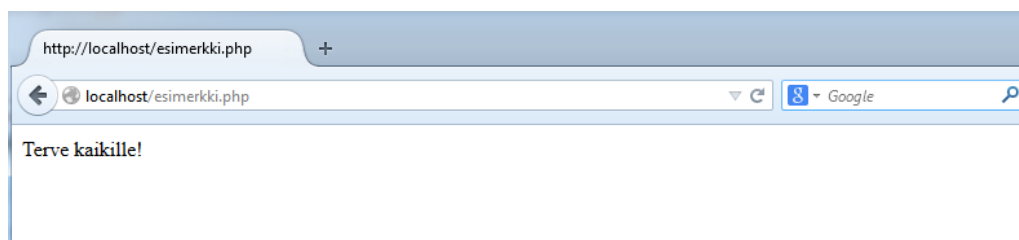
PHP on komentosarjakieli, jota käytetään yleisesti web-palveluiden tuottamiseen. Se on helposti opittavissa ja sen syntaksi muistuttaa C-, Java- ja Perl-kieliä. PHP-sivujen sisältämä koodi suoritetaan palvelinpuolella oman tulkin avulla ennen kuin sivu ladataan käyttäjän selaimen. PHP-koodia voidaan myös sisällyttää HTML-dokumentteihin PHP-tagien (`<?php ?>`) avulla. Kuvassa 5 on esimerkki PHP-tiedostosta, jossa määritellään yksinkertainen tervehdys-funktio ja kutsutaan sitä. Esimerkki.php -tiedoston ajotulokset nähdään kuvassa 6. (PHP 2014.)

```

esimerkki.php
1  <?php
2  // Luodaan funktio tervehdystä varten
3  function tervehdys(){
4      echo "Terve kaikille!";
5  }
6
7  tervehdys(); // Kutsutaan tervehdys-funktiota
8
9  ?>

```

KUVA 5. Esimerkki PHP -tiedostosta



KUVA 6. Esimerkki.php -tiedoston tulokset selaimessa



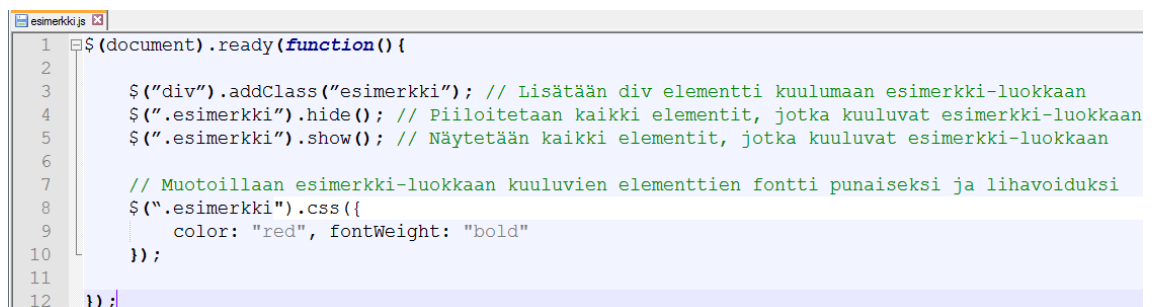
MySQL-tietokantayhteyksiin käytettiin PDO-rajapintaa, joka on ollut mukana PHP:ssä versiosta 5.1 alkaen. PDO:n avulla voidaan parantaa tietoturvaa esikäsittelemällä muuttajat, joita käytetään tietokantakyselyissä.

## 4.4 JavaScript-kieli

JavaScript on oliopohjainen skriptikieli, jolla yleensä lisätään interaktiivisuutta ja dynaamisuutta www-sivustoille. JavaScriptin avulla voidaan varmistaa, että käyttäjän syötteet ovat oikeassa muodossa lomaketta täytettäessä. Esimerkiksi, jos käyttäjä kirjoittaa tekstiä numeroiden sijaan ikää kysyttäessä niin virheilmoitus voidaan antaa ilman, että koko sivua tarvitsee ladata uudelleen. JavaScriptiä voidaan myös hyödyntää ulkoasua koskevissa ongelmissa. Esimerkiksi, jos alasetovalikon koko kasvaa niin suureksi, että se aiheuttaa päällekkäisyyksiä jonkin toisen elementin kanssa. Tämä voidaan korjata määrittämällä alasetovalikon mitat JavaScriptin avulla ja siirtämällä elementtejä dynaamisesti tätä tietoa hyödyntäen.

### 4.4.1 JQuery-kirjasto

jQuery on JavaScriptin kirjasto, joka tuotiin mukaan projektiin helpottamaan AJAX-kutsujen käyttöä. jQuery yksinkertaistaa JavaScriptin käyttöä ja helpottaa mm. elementtien valitsemista. jQuery tukee myös yleisimpiä selaimia, jonka ansiosta toteutukset saadaan toimimaan helpommin myös Internet Explorerilla. Kuvassa 7 on esitetty esimerkkejä jQueryn käytöstä kommentteineen. Esimerkit ovat sisällytetty \$(document).ready() -metodiin, joka varmistaa, että web-sivun lataaminen on suoritettu loppuun ennen kuin jQuery-koodeja suoritetaan selaimessa.



```

1 $(document).ready(function() {
2
3     $("div").addClass("esimerkki"); // Lisätään div elementti kuulumaan esimerkki-luokkaan
4     $(".esimerkki").hide(); // Piiloitetaan kaikki elementit, jotka kuuluvat esimerkki-luokkaan
5     $(".esimerkki").show(); // Näytetään kaikki elementit, jotka kuuluvat esimerkki-luokkaan
6
7     // Muotoillaan esimerkki-luokkaan kuuluvien elementtien fontti punaiseksi ja lihavoiduksi
8     $(".esimerkki").css({
9         color: "red", fontWeight: "bold"
10    });
11
12 });

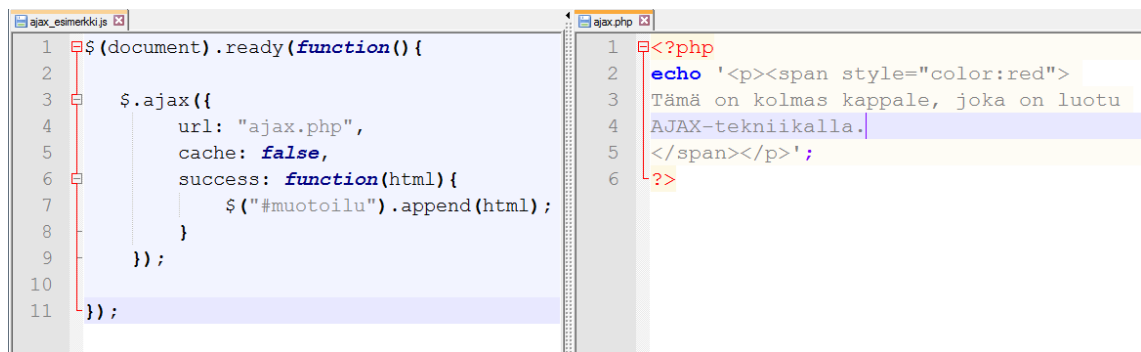
```

KUVA 7. Esimerkkejä jQueryn käytöstä

## 4.4.2 AJAX-tekniikka

AJAX-tekniikka mahdollistaa reaaliaikaisen interaktiivisuuden web-sivustoilla. AJAX-kutsuilla voidaan päivittää osia web-sivusta ilman, että koko sivua tarvitsee ladata uudelleen. Yksi ensimmäisistä ja tunnetuimmista AJAX-sovelluksista on Googlen hakupalvelu, joka ehdottaa hakusanoja käyttäjän syötteiden perusteella.

Kuvassa 8 on yksinkertainen esimerkki AJAX-kutsun käytöstä. Kuvassa nähdään kahden tiedoston sisältämät koodit, joilla lisätään CSS:n esimerkkiin (KUVA 3) kolmas tekstikappale. Yleensä AJAX -kutsut liitetään johonkin tapahtumankäsittelijään, kuten napin painallukseen. Tässä yksinkertaisessa esimerkissä AJAX-kutsu on sisällytetty kuitenkin `$(document).ready()` -metodiin, joten se suoritetaan heti web-sivun latauduttua.



```

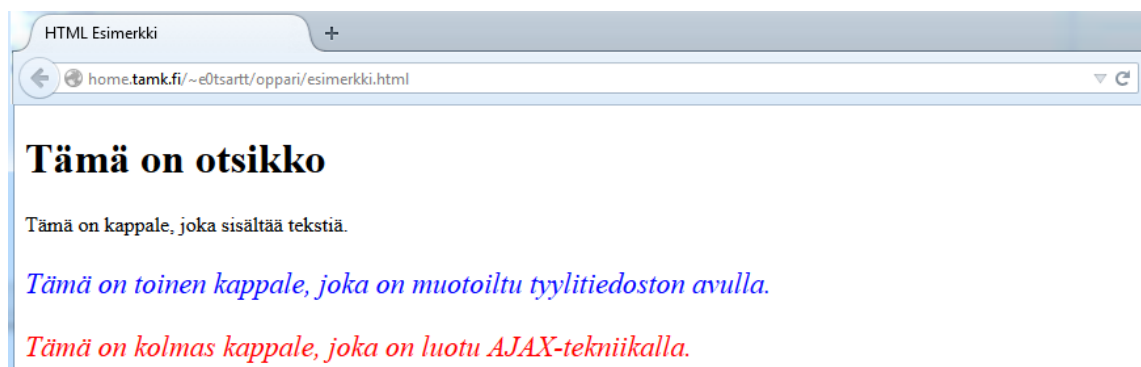
ajax_esimerkki.js
1 $(document).ready(function() {
2
3     $.ajax({
4         url: "ajax.php",
5         cache: false,
6         success: function(html) {
7             $("#muotoilu").append(html);
8         }
9     });
10
11 });

ajax.php
1 <?php
2 echo '<p><span style="color:red">
3 Tämä on kolmas kappale, joka on luotu
4 AJAX-tekniikalla.
5 </span></p>';
6 <?>

```

KUVA 8. Esimerkki AJAX-kutsun käytöstä

Edellisen esimerkin tulokset nähdään Kuvassa 9, johon on ilmestynyt punaisella fontilla kolmas tekstikappale. Tulosta voi verrata kuvaan 4.



KUVA 9. AJAX-kutsun tulokset selaimessa

Suurin osa Nurseus-järjestelmän toiminnoista siirrettiin käyttämään AJAX-tekniikkaa jatkokehityksen aikana ja se vei kaikkein eniten aikaa tässä projektissa. Tämän ansiosta järjestelmästä saatiin käyttäjäystävällisempi ja nopeampi käyttää, koska sivuja ei tarvitse aina ladata uudelleen toimintojen jälkeen.

#### 4.5 SQL-kyselykieli ja MySQL-tietokanta

MySQL on avoimen lähdekoodin relaatiotietokantaohjelmisto. Relaatiotietokannassa tiedot tallennetaan tauluihin ja tietojen suhteet eli relaatiot kuvaavat miten tiedot liittyvät toisiinsa.

Relaatiotietokantoja hallinnoidaan IBM:n kehittämällä SQL-kyselykielellä. SQL:n perustana ovat yksinkertaiset CRUD-operaatiot (create, retrieve, update, delete). Taulukossa 5 on kuvattu CRUD-operaatioiden syntaksi SQL-kyselykielellä.

TAULUKKO 5. SQL-syntaksi

Operaatio	SQL
Create	INSERT
Retrieve	SELECT
Update	UPDATE
Delete	DELETE

#### 4.6 Notepad++-tekstieditori

Notepad++ on avoimeen lähdekoodiin perustuva tekstieditori, joka on suunniteltu ohjelmoinnin työvälineeksi windows-ympäristöön. Notepad++:in ominaisuuksia ovat mm. syntaksiväritys useille ohjelmointikielille, hyvät etsi ja korvaa -ominaisuudet, tukee välilehtiä sekä kahden tiedoston samanaikaista katselua (Multi-View). Näiden ominaisuuksien sekä aikaisemman kokemuksen perusteella Notepad++ valittiin työkaluksi tähän projektiin. (Notepad++ features 2014.)

## 4.7 Firebug-liitännäinen

Firebug on Mozilla Firefox-selaimen liitännäinen, joka tarjoaa paljon työkaluja web-kehittäjän avuksi. Firebugin inspect element -toiminnolla voi tarkastella sivustojen lähdekoodia monipuolisesti. CSS -paneeli näyttää valittuun elementtiin vaikuttavat tyyli-määräykset. Firebug mahdollistaa myös HTML:n ja CSS:n muokkauksen. Muokkaus tapahtuu suoraan HTML- ja CSS -paneelisiin ja muutokset näkyvät heti selaimessa. Firebugilla tehdyt muutokset eivät tallennu mihinkään, joten sillä on hyvä tehdä kokei-luja kehitystä varten. Tehdyt muutokset häviävät kun sivu ladataan uudelleen.

(What is Firebug? 2014.)

Firebug sisältää hyviä apuvälineitä myös JavaScript-kieltä varten. Console -välilehti näyttää sivulla tapahtuvat JavaScript-virheet. Virheilmoituksia klikkaamalla nähdään virheen sijainti koodissa. AJAX -kutsut näkyvät Net -välilehdellä ja kutsuista nähdään esimerkiksi mihin osoitteeseen kutsu tehtiin, minkälaisin parametrein ja mikä vastaus oli. Kuvassa 10 tarkastellaan AJAX -kutsua Firebugin avulla. Kuvassa oleva AJAX-kutsu tapahtuu hoidon tarpeen määrittämisen yhteydessä ja kuvasta voidaan nähdä, että lomakkeella oleva tieto välittyy selaimesta web-palvelimelle POST-metodilla. Web-palvelin suorittaa PHP-tulkin avulla student\_work\_actions.php -tiedoston sisältämän skriptin lomakkeelta välitetyillä parametreilla.

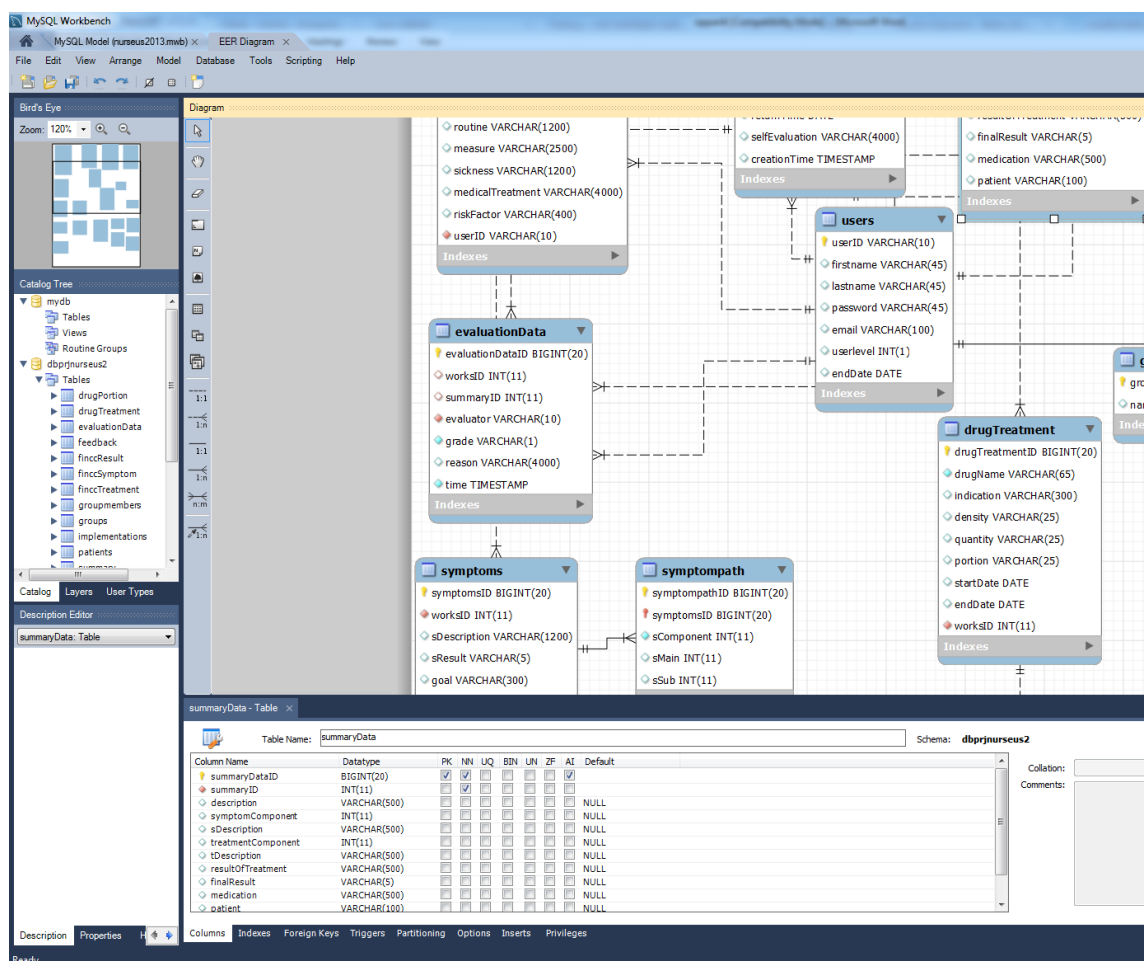
The screenshot displays a web application interface and the Firebug developer tool. The web page shows a form for "Hoidon tarpeen määrittäminen" (Setting the need for treatment). The form includes a "Päivämäärä" (Date) field set to "21.05.2014" and a "Kuume" (Fever) field. Below the form is a table for "Tarve" (Need) with columns for "Komponentti" (Component), "Paaluokka" (Room), "Alaluokka" (Sub-room), and "Kirjoita kuvaus: (max. 1100 merkkiä) merkkejä jäljellä: 1047". A "Valitse komponentti" dropdown menu is visible. The Firebug console shows a list of network requests, with "POST student\_work\_actions.php" selected. The response pane shows the data sent to the server, including "symptomDescription" with the value "Kuumetta ollut kotona kolmena päivänä yli 39 astetta.".

KUVA 10. AJAX-kutsun tarkastelu Firebug-liitännäisen avulla

## 4.8 MySQL Workbench -ohjelma

MySQL Workbench on työkalu tietokannan suunnitteluun ja hallintaan. Graafisen käyttöliittymän avulla voidaan tietokannasta luoda ER-kaavio ja ER-kaavion avulla voidaan luoda tietokanta MySQL-palvelimelle. Tietokannan tietoja voidaan myös muokata, poistaa sekä niistä voidaan ottaa varmuuskopioita Workbenchin avulla. Kuvassa 11 on mallinnettu Nurseus-järjestelmän tietokantaa MySQL Workbenchin avulla ja liitteessä 2 on nähtävissä Nurseus-järjestelmän ER-kaavio.

(MySQL Workbench 2014.)



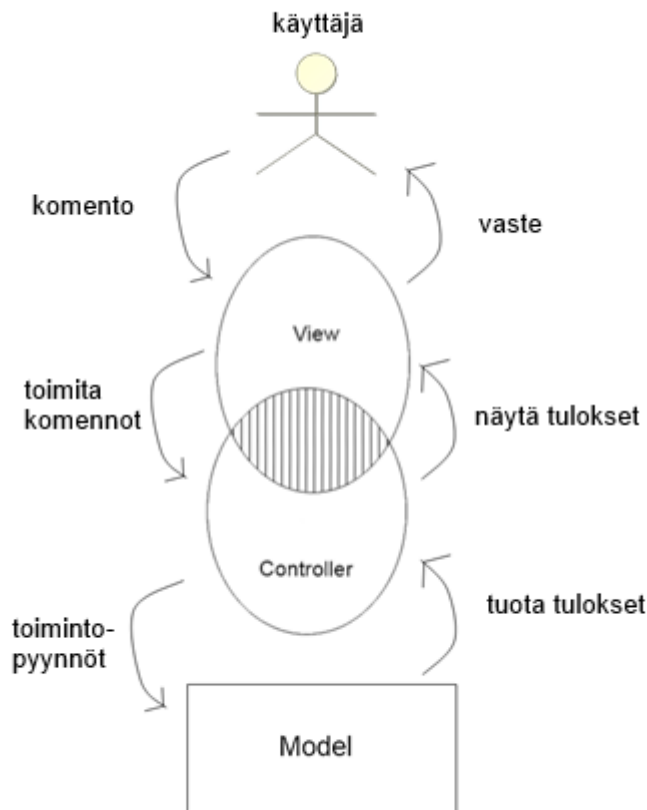
KUVA 11. Tietokannan suunnittelua MySQL Workbench 6.1 -työkalun avulla

## 5 PÄIVITETTY NURSEUS-JÄRJESTELMÄ

Tässä luvussa käydään läpi Nurseus-järjestelmään jatkokehityksen aikana tehdyt päivitykset sekä kuvataan järjestelmän arkkitehtuuri.

### 5.1 Nurseus-järjestelmän arkkitehtuuri

Järjestelmän kehityksessä on käytetty osittaisen MVC-mallin mukaista arkkitehtuuria. Järjestelmän tietokanta on kokonaan omassa kerroksessa ja näkymä sekä ohjain on liittämättä. Kuvassa 12 on kuvattu järjestelmän arkkitehtuuri sekä käyttäjän vuorovaikutus järjestelmän kanssa.



KUVA 12. Nurseus-järjestelmän osittainen MVC-arkkitehtuuri

### 5.2 Välilehtinäköymä

Jatkokehityksen aikana Nurseus-järjestelmään luotiin välilehtinäköymä (KUVA 13) hoitosuunnitelman tekoa varten. Välilehtinäköymässä potilaan tiedot, hoidon suunnittelu,

toteutus, lääkehoito sekä yhteenveto ovat erillisinä välilehtinä. Tämä selkeyttää hoitosuunnitelman tekoa, kun asiakokonaisuudet on eritelty toisistaan. Välilehtinäkömää toteutettiin jQuery UI:n avulla. Kaikki toiminnallisuus välilehdillä tehtiin dynaamiseksi AJAX-kutsuja käyttäen. Tämä mahdollistaa eri välilehtien selaamisen ilman, että käyttäjän syötteet katoaisivat lomakkeiden kentistä.

Olet kirjautuneena käyttäjänä:  
Testi Opiskelija  
Käyttäjätaso: Opiskelija  
Kirjautu ulos / Log out  
Torstai 20.3.2014 klo 9:52, aikakatkaisu: 10:19

**Nurseus**

**Menu**  
Etusivu  
Muokkaa omia tietoja  
Ryhmähallinta  
Työt  
Ohjeet

**Linkkejä**  
FinCC käyttöopas 3.0. Luokitukset liitteessä.  
Hoidata -loppuraportti. Hoitosuunnitelman arviointilomake liite 2.  
Potilastiedon arkiston verkkokoulu.  
Tietoturvakurssi.  
Potilastiedon arkiston käyttöönoton käsikirja.  
Ydintiedot\_otsikot ja näkymät -opas\_versio 3.0.

**Anna järjestelmästä palautetta:**  
Palaute

**Hoitosuunnitelma**

Potilaan tiedot | **Hoidon suunnittelu** | Toteutus | Lääkehoito | Yhteenveto

Lisää tarve Takaisin työsuivulle

PALAUTA hoitosuunnitelma POISTA hoitosuunnitelma

**Hoidon suunnittelu:**

**Hoidon tarve** 20.02.2021

- Nestetasapaino => Muuttunut nestevolyymi => Kuivuminen  
Perifeeriset suonet näkymättömissä, iho nousee poimulle. Kieli kuiva. Verenpaine matala/

**Hoidon tavoite**  
Saa nesteitä, 1 vuorokaudessa

**Suunnitellut toiminnot**

- Nestetasapaino => Nestetasapainon seuranta  
Nestelista
- Ravitsemus => Ravitsemustilan seuranta => Painon kontrollointi päivittäin
- Nestetasapaino => Nesteytyksestä huolehtiminen => Suonensisäinen nesteytys tarvittaessa korvaus NaCl

Muokkaa tarvetta, tavoitetta ja toimintoa Poista

**Hoidon tarve** 20.02.2021

- Ravitsemus => Ravitsemuksen muutos => Ruokahaluttomuus  
Ruokahalu ollut huono pitkään. Syö yksipuolisesti lähinnä eineksiä. Laihtunut 8kg viime syyskuusta.

**Hoidon tavoite**  
Saa riittävästi energiaa (1400kcal).  
Syö vähintään puolet aterioistaan

**Suunnitellut toiminnot**

- Ravitsemus => Ravinnon saannin turvaaminen => Ruokailun seuranta  
Arvioidaan loppuilla eväty määrä

KUVA 13. Hoitosuunnitelman välilehtinäkömää

**Potilaan tiedot** -välilehdelle kirjataan potilaan yksilöintitietoja kuten nimi ja ikä sekä potilaan hoitoon liittyviä esitietoja. Esitiedot koostuvat hoitoon hakeutumisen syystä, terveystottumuksista, fysiologisista mittauksista, aikaisemmista sairauksista, lääkehoitoon liittyvistä tiedoista sekä potilaan terveyteen liittyvistä riskitekijöistä.

**Hoidon suunnittelu** -välilehti sisältää hoidon tarpeen, tavoitteen sekä suunnitellut toiminnot. Tarpeita sekä toimintoja kirjattaessa käyttäjän pitää lisätä myös FinCC-luokitukset, jotka näkyvät kuvassa vihreällä.

**Toteutus**-välilehdellä jokaiselle suunnitellulle toiminnolle voi kirjata toteutuksen. Toteutus sisältää hoidon tavoitteiden saavuttamiseksi valitut keinot ja menetelmät.

**Lääkehoito**-välilehdelle kirjataan lääkitykseen liittyviä tietoja kuten lääkkeen nimi, vahvuus ja käyttötarkoitus sekä annosteluun liittyvät tiedot.

**Yhteenveto**-välilehdelle kirjataan hoitajakson yhteenveto. Yhteenveto sisältää mm. hoidon tarpeen, hoitajakson aikana tehdyt toiminnot ja hoidon tulokset.

### 5.3 Hakutoiminto

Järjestelmään toteutettiin hakutoiminto automaattisella tekstintäydennyksellä FinCC-luokituksia varten. Kun hakukenttään kirjoitetaan enemmän kuin kaksi merkkiä, niin järjestelmän automaattinen tekstintäydennys ehdottaa luokituksia syötteen perusteella. Lista päivittyy koko ajan kirjoittaessa, eli mitä enemmän merkkejä sitä tarkempi hakutulos (KUVA 14.).

Hakutuloksen valinnan jälkeen järjestelmä lisää automaattisesti luokitukset niille varattuihin alasvetovalikoihin. Esimerkiksi haettaessa alaluokkaa hoidon tarpeelle, niin järjestelmä lisää myös siihen kuuluvan pääluokan ja komponenttitason alasvetovalikoihin.

The screenshot displays the 'Hoidon tarpeen määrittäminen' (Setting care needs) interface in the Nurseus system. The page has an orange header with the 'Nurseus' logo. In the top right corner, it shows the user is logged in as 'Testi Opiskelija' (Test Student) with the role 'Opiskelija' (Student) and the date/time 'Torstai 20.3.2014 klo 10:06, aikakatkaisu: 10:36'. On the left, there is a 'Menu' section with links like 'Etusivu', 'Muokkaa omia tietoja', 'Ryhmähallinta', 'Työt', and 'Ohjeet', and a 'Linkkejä' (Links) section with various document links. The main content area is titled 'Hoidon tarpeen määrittäminen' and includes a search bar with the text 'ver' and a 'Hae' button. Below the search bar, a list of suggestions is shown, including 'Verivirtsaisuus', 'Verenvuoto', 'Verenkierto', 'Verenkiertohäiriö', 'Verenpaineen muutos', 'Verenkiertoon liittyvä tiedon puute', 'Verensokerin muutos', 'Ääreisverenkierron muutos', and 'Veriuloste'. A 'Tallenna työhön: Nurseusinfo' button is visible. At the bottom right of the suggestions list, there is a 'Poista' button and a note '(max. 1100 merkkiä) merkkejä jäljellä'. The footer contains the text 'Copyright © TAMK'.

KUVA 14. Hakutoiminnon automaattinen tekstintäydennys



## 5.4 Lääkehoito

Järjestelmään toteutettiin uutena komponenttina lääkehoidon lomake, johon kirjataan potilaalle määrätty lääkkeet ja niihin liittyvät tiedot. Tämän toteuttamiseksi oli MySQL-tietokantaan lisättävä tarvittavat taulut sekä muokattava hoitosuunnitelmaan liittyviä PHP -tiedostoja.

KUVA 15. Lääkehoidon lomake

## 5.5 Yhteenvetodon muutokset

Jatkokehityksen aikana järjestelmään lisättiin FinCC-rakenteisen yhteenvetolomakkeen rinnalle ydintietotasoinen yhteenvetolomake (KUVA 16), joka on vaihtoehtona FinCC-rakenteiselle lomakkeelle. Ydintietotasoinen lomake on vapaamuotoinen lomake, johon kirjataan hoitojakson yhteenveto ilman FinCC-luokituksia. Asiakas halusi tämän järjestelmään lisäominaisuutena, jota voidaan käyttää erityistapauksissa. Pääsääntöisesti järjestelmää käytetään kuitenkin FinCC-luokitusten opiskelemiseen.

Potilaan tiedot	Hoidon suunnittelu	Toteutus	Lääkehoito	Yhteenveto
<b>Potilas</b>				
Nimi ja ikä	esim. Matti Meikäläinen (60 vuotta)			
<b>Hoidon arviointi</b>				
<b>Loppuarvio:</b> (max. 1200 merkkiä) merkkejä jäljellä:				
<b>Hoidon tarve:</b> (max. 1200 merkkiä) merkkejä jäljellä:				
<b>Hoitotyön toiminnot:</b> (max. 1200 merkkiä) merkkejä jäljellä:				
<b>Hoidon tulokset:</b> (max. 1200 merkkiä) merkkejä jäljellä:				
<b>Lääkehoito:</b> (max. 1200 merkkiä) merkkejä jäljellä:				
tallenna				
Copyright © TAMK				

KUVA 16. Ydintietotasoinen yhteenvetolomake

FinCC-rakenteiseen yhteenvetolomakkeeseen lisättiin uutena ominaisuutena automaattinen tietojen haku tietokannasta. Lomakkeen kenttiin haetaan pohjatiedot hoitosuunnitelman tietojen perusteella ja käyttäjän tarvitsee vain muokata näitä mieleisikseen. Kuvassa 17 nähdään, että ”Kuumetta 39 viikon ajan” on tuotu hoitosuunnitelmasta verenkierto -kenttään. Lomakkeen kenttiin lisättiin myös merkkilaskuri, joka nähdään kuvassa kenttien oikealla puolella.

Yhteenvedon tiedot jaettiin asiakokonaisuuksiin jQuery UI Accordion -pienoisohjelmaa hyödyntäen. Asiakokonaisuuden voi piilottaa klikkaamalla otsikkoa hiirellä ja näin olleen sivun koko ei kasva liian suureksi. Tästä on esimerkki kuvassa 17, jossa loppuarvio sekä hoitotyön toiminnot ovat piilotettuna.

**Menu**

[Etusivu](#)  
[Muokkaa omia tietoja](#)  
[Ryhmähallinta](#)  
[Työt](#)  
[Ohjeet](#)

**Linkejä**

[FinCC käyttöopas 3.0, Luokitukset liitteessä](#)  
[Hoidata -loppuraportti, Hoitosuunnitelman arviointilomake liite 2](#)  
[Pottitiedon arkiston verkkokoulu](#)  
[Tietoturvakurssi](#)  
[Pottitiedon arkiston käyttöönoton käsikirja](#)  
[Ydintiedot, otsikot ja näymät -opas, versio 3.0](#)

**Anna järjestelmästä palautetta:**  
[Palaute](#)

## Hoitosuunnitelma

Potilaan tiedot
Hoidon suunnittelu
Toteutus
Lääkehoito
Yhteenvedo

Tehdään työstä yhteenvedo...  
 Lisättiin 2 kohde(tta) yhteenvedoan

**Yhteenvedo**

Yhteenvedon nimi:	testi 10.12(yhteenvedo)	
Potilaan nimi ja ikä:	Heikki (63 vuotta)	esim. Matti Meikalainen (60 vuotta)
Alkuperäinen:	Heikki	

▶ **Loppuarvio**

▼ **Hoidontarve**

**Verenkierto** Kuumetta 39 viikon ajan. 24/1200

**Hengitys** Kuukauden verran ahdistusta, joka pahenee rasituksessa 54/1200

Aktiviteetti

▶ **Hoitotyön toiminnot**

▼ **Hoidon tulokset**

Aktiviteetti  0/1200

-valitse tulosluokitus-

KUVA 17. Hoitosuunnitelman yhteenvedo

## 5.6 Aikakatkaaisu

Järjestelmän alkuperäinen aikakatkaaisu oli toteutettu sillä tavoin, että istunnon aikakatkaaisu tapahtui 30 minuutin kuluessa vaikka käyttäjä käyttäisi järjestelmää aktiivisesti istunnon aikana. Jatkokehityksen aikana aikakatkaistua muutettiin, jotta käyttäjän istunto kestää niin kauan kunnes käyttäjä kirjautuu itse ulos tai on käyttämättä järjestelmää 30 minuutin ajan. Aikakatkaistuksen toteutuksessa hyödynnettiin PHP:n istuntomuuttujia sekä JavaScriptiä.

## 5.7 Käyttö- ja asennusohjeet

Jatkokehityksen toteutuksen yhteydessä Nurseus -järjestelmälle laadittiin 20 sivun käyttöohjeet, jotka sisällytettiin järjestelmään. Järjestelmälle toteutettiin myös 7 sivun asennusohjeet tulevaisuutta varten, jos järjestelmä mahdollisesti myydään muille oppilaitoksille.

## 6 YHTEENVETO JA NUSEUS-JÄRJESTELMÄN TULEVAISUUS

Nurseus-järjestelmän jatkokehitys oli projektina onnistunut. Asiakkaan mielestä jatkokehitykselle asetetut tavoitteet ylitettiin ja kehittäjän näkökulmasta projekti oli palkitseva kartutettujen taitojen muodossa. TAMK:n tietotekniikan koulutusohjelmaan ei web-ohjelmointi kuulunut, joten kaikki tarvittavat tiedot Nurseus-järjestelmän kehitykseen piti hankkia itse.

Interaktiivinen Tekniikka Koulutuksessa (ITK) on Suomen suurin tieto- ja viestintätekniiikan opetuskäytön konferenssi. Vuonna 2014 ITK järjestettiin Hämeenlinnassa, jossa Nurseus oli esiteltävänä TAMK:n lehtorien Taru Lehtimäen ja Helena Vesaluoman toimesta. ITK2014:n jälkeen Nurseus on herättänyt kiinnostusta eri oppilaitoksissa ja muutamia yhteydenottoja mahdollisesta järjestelmän ostamisesta on tullut tämän dokumentin kirjoitushetkellä. (ITK 2014.)

Jatkokehitykselle on edelleen tarvetta, koska oppimisympäristönä Nurseus-järjestelmän kaikkia mahdollisuuksia ei ole vielä käytetty ja lisäominaisuuksien tarpeita tulee jatkuvasti lisää. Jatkokehitystä ja järjestelmän ylläpitoa tarvitaan myös varmistamaan se, että Nurseus onnistuisi kaupallisena tuotteena.

Tietotekniikan Tuotekehitys ja Innovointi (TKI) Verstas on tietotekniikan opiskelijoiden perustama osuuskunta. TKI-Verstaan toimintaperiaate on etsiä toimeksiantoon sopivat ehdokkaat pääsääntöisesti TAMK:n opiskelijoiden keskuudesta. Toimintaympäristönä on ohjelmistotekniikka, sulautetut järjestelmät ja elektroniikka sekä tietoliikennetekniikka ja tietoverkot. (TKI-Verstas 2014.)

Tämän dokumentin kirjoitushetkellä sopimusneuvottelut ovat käynnissä TAMK:n ja TKI-verstas Osk:n välillä. Tarkoituksena on siirtää vastuu Nurseus-järjestelmän jatkokehityksestä ja järjestelmän tuotteistamisesta Tki-verstas Osk:lle.

## LÄHTEET

Firebug. What is Firebug? 2014. Luettu 15.5.2014.

<https://getfirebug.com/whatisfirebug>

Haikala, I. & Mikkonen, T. 2011. Ohjelmistotuotannon käytännöt. Helsinki: Talentum Media Oy.

ITK. 2014. Luettu 20.5.2014.

<http://www.itk.fi/2014/ohjelma/foorumi/536>

Liljamo, P., Kinnunen, U. & Ensio, A. 2012. FinCC-luokituskokonaisuuden käyttöopas - SHTaL 3.0, SHToL 3.0, SHTuL 1.0. Helsinki: Terveyden ja hyvinvoinnin laitos.

Lunn, I. 2012. CSS3 Foundations. United Kingdom: John Wiley & sons Ltd.

Matthew, W. 2012. HTML5 Foundations. United Kingdom: John Wiley & sons Ltd.

MySQL Workbench. 2014. Luettu 25.3.2014.

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

Notepad++. Notepad++ features. 2014. Luettu 15.5.2014.

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

Nurseusinfo. 2014. Nurseus-järjestelmän info-sivu. Luettu 11.5.2014.

<http://nurseusinfo.tamk.fi/>

PHP. 2014. Luettu 15.3.2014.

<http://php.net/>

## LIITTEET

Liite 1. Aiheeseen liittyviä linkkejä.

### **Ketterät menetelmät:**

<http://agilemanifesto.org/>

<http://www.scrumalliance.org/community/articles/2013/2013-april/what-does-the-agile-manifesto-mean>

### **HTML:**

<http://html.net/tutorials/html/>

<http://www.w3schools.com/html/DEFAULT.asp>

### **CSS:**

<http://html.net/tutorials/css/>

<http://www.csszengarden.com/>

### **PHP & MySQL**

<http://www.freewebmasterhelp.com/tutorials/phpmysql>

### **PHP PDO:**

<http://code.tutsplus.com/tutorials/php-database-access-are-you-doing-it-correctly--net-25338>

### **JavaScript ja jQuery**

<http://www.learn-javascript-tutorial.com/>

<http://api.jquery.com/>

<http://www.jquery-tutorial.net/>

### **AJAX-kutsut jQueryn avulla:**

<http://code.tutsplus.com/tutorials/5-ways-to-make-ajax-calls-with-jquery--net-6289>

### **jQuery ui:**

<http://jqueryui.com/>

## Liite 2. Nurseus-järjestelmän ER-kaavio

