

# KEMI-TORNION AMMATTIKORKEAKOULU

## Opetussuunnitelmasovellus Linux/Qt

Rasila Kimmo

Tietotekniikan koulutusohjelman opinnäytetyö  
Ohjelmistotekniikka  
Insinööri(AMK)

KEMI 2012

## ALKUSANAT

Aluksi haluan kiittää entistä vastuukouluttajaani Marko Heikkilää tästä kyseisestä projektista. Projektin laajentuessa, päätimme yhdessä että tästä tulisi minun opinnäytetyöni.

Haluan kiittää myös opinnäytetyöni ohjaajaa Tapani Ruokasta käytännön ohjauksesta sekä hyvistä neuvoista ja kärsivällisyydestä opinnäytetyön kirjoitusprosessin aikana.

Osoitan kiitokseni myös Kaikkosen Lailalle, jonka kärsivällisyydestä tiettyjen kurssien kohdalla olen todella hyvilläni. Lisäksi kiitän myös lukuisia muita Kemi-Tornion ammatti-korkeakoulun opettajia, joiden korkeatasoisen opetuksen johdosta minulla on kehittynyt riittävä osaaminen tämän kyseisen projektin toteuttamiselle.

Haluan kiittää myös Pekka Harjuniemeä sekä Teppo Aaltoa, projektin alkuvaikeuksien aikana saamistani käytännön neuvoista ja tuesta.

Kiitän kaikkia läheisiäni ja ystäviäni kannustuksesta sekä tuesta opintojeni aikana. Erityiskiitos parhaalle ystävälleni Veijo Väänäselle, opintojeni alussa saamastani tuesta sekä rakkaalle vaimolleni Tiina Koskela-Rasilalle. Ilman hänen tukeaan, en olisi näitä opintoja kyennyt toteuttamaan.

Kempeleessä 23.4.2012

Kimmo

## TIIVISTELMÄ

Kemi-Tornion ammattikorkeakoulu,	Tekniikan ala
Koulutusohjelma	Tietotekniikka
Opinnäytetyön tekijä	Kimmo Rasila
Opinnäytetyön nimi	Opetussuunnitelmasovellus
Työn laji	Opinnäytetyö
Päiväys	23.4.2012
Sivumäärä	42 sivua
Opinnäytetyön ohjaaja	Tapani Ruokanen, Yliopettaja

Tämä opinnäytetyö tehtiin Kemi-Tornion ammattikorkeakoulun toimeksiantona. Opinnäytetyön tarkoituksena oli suunnitella ja toteuttaa mahdollisimman monipuolisin ominaisuuksin varustettu, silti pelkistetyin yksinkertainen, helppokäyttöinen ja toimiva graafinen käyttöliittymä Kemi-Tornion ammattikorkeakoulun eri opetussuunnitelmatietojen päivittämisen sekä ylläpidon helpottamiseksi.

Opinnäytetyössä tutkittiin mahdollisuuksia toteuttaa sovellus ja sen graafinen käyttöliittymä Qt-ohjelmointiympäristössä C++-kielellä, koska tämä tarjoaa varsin hyvät puitteet näyttävien graafisten käyttöliittymien toteutukseen. Opinnäytetyöhön liittyvä verkkosivusto päätettiin toteuttaa PHP-kielellä sen tarjoamien tietokantafunktioiden johdosta. Edellä mainittuihin ohjelmointikieliin päädyttiin muun muassa sekä alustariippumattomuuden että sovelluksen kehittämistä koskevien pohdintojen seurauksena.

Opinnäytetyön graafisen käyttöliittymän sekä tietokannan suunnittelussa käytettiin apuna Microsoft Visiota. Käyttöliittymä tehtiin Qt-ohjelmointiympäristössä C++-kielellä. Tietokanta toteutettiin MySQL-tietokannalla ja verkkosivusto PHP-kielellä.

Opinnäytetyön lopputuloksena saatiin alkuperäisten tavoitteiden mukainen käyttöliittymä, joka monipuolisista toiminnoistaan huolimatta saatiin rakennettua sellaiseksi, että toiminnallinen kokonaisuus on varsin yksinkertainen sekä selkeän helppokäyttöinen.

Asiasanat: graafisen käyttöliittymän toteutus, QtSDK, C++, PHP.

## ABSTRACT

Kemi-Tornio University of Applied Sciences, Technology	
Degree Programme	Information Technology
Name	Kimmo Rasila
Title	Curriculum software
Type of Study	Bachelor's Thesis
Date	23 April 2012
Pages	42 pages
Instructor	Tapani Ruokanen, Principal lecturer

This work was commissioned by Kemi-Tornio University of Applied Sciences. Article of this thesis was to design and implement the most feature-packed, yet minimalist simple, easy to use and the most functional graphical user interface for to updating and maintaining of the students curriculums of the Kemi-Tornio University of Applied Sciences.

This thesis investigates the feasibility of creating an application, and graphical user interface in Qt-programming environment in C++-language, as this provides a fairly good looking framework for the implementation of graphical user interfaces. Website associated this thesis was implemented in PHP, mostly because of the database functions offered by the PHP-language. The above mentioned programming languages were chosen, mostly because of the independence of the environment and discussions about the development.

The user interface of the thesis as well as the design of the database was made with Microsoft Visio. User interface was made in Qt-programming environment with C++. The database was created in MySQL-database and the website was made with PHP-language.

The end result of this thesis was the user interface of the original objectives. In spite of the wide range of activities, functionality of this program is fairly simple and easy to use.

Keywords: creating a Graphical User Interface, QtSDK, C++, PHP.

## SISÄLLYSLUETTELO

ALKUSANAT .....	1
TIIVISTELMÄ .....	2
ABSTRACT .....	3
SISÄLLYSLUETTELO .....	4
1. JOHDANTO .....	5
2. TAVOITTEET .....	6
3. TYÖVÄLINEET .....	7
3.1. QT SDK .....	7
3.1.1. Qt:n historia .....	7
3.1.2. Qt Creator .....	8
3.1.3. Signal-slot-mekanismi .....	9
Qt:n lisensioinnit .....	10
Qt-moduulit .....	10
3.2. PHP .....	10
3.2.1. Historia .....	10
3.2.2. PHP koodi .....	11
3.2.3. Open Source-tuote .....	13
3.2.4. PHP siirrettävyys .....	13
3.3. MYSQL .....	13
3.3.1. Historia .....	13
3.3.2. phpMyAdmin .....	14
3.4. LINUX .....	15
3.4.1. Historia .....	15
3.4.2. Linux-kernel .....	16
3.4.3. Komentotulkki .....	17
4. OHJELMISTON SUUNNITTELU .....	19
4.1. Suunnittelumenetelmät .....	19
4.2. Käyttöliittymän suunnittelu .....	22
4.3. Tietokannan suunnittelu .....	23
4.4. Verkkosivuston suunnittelu .....	24
5. OHJELMISTON TOTEUTUS .....	25
5.1. Käyttöliittymän toteutus .....	25
5.2. Tietokannan toteutus .....	30
5.3. Verkkosivuston toteutus .....	32
6. OHJELMISTON TESTAUS .....	35
6.1. TestLink .....	35
6.2. BugZilla .....	36
6.3. Ohjelmistokokonaisuuden testaus .....	37
7. POHDINTA/JATKOKEHITYSMAHDOLLISUUDET .....	38
8. YHTEENVETO .....	40
9. LÄHDELUETTELO .....	41

## 1. JOHDANTO

Kemi-Tornion ammattikorkeakoulu on Suomen toiseksi pohjoisin ammattikorkeakouluyksikkö. Sen kirjoilla opiskelee noin 2400 oppilasta, kolmessakymmenessä eri koulutusohjelmassa.

Tällä hetkellä eri koulutusohjelmien opetussuunnitelmat ladataan koulun verkkosivustolle HTML- sekä JavaScript-tiedostoina, minkä vuoksi tietojen päivittäminen on pelkästään sivuston ylläpitäjän taholta toteutettavissa.

Opinnäytetyön tarkoituksena oli suunnitella ja toteuttaa Qt-ohjelmointiympäristöä, C++ -, PHP-ohjelmointia sekä MySQL-tietokantoja apuna käyttäen graafinen käyttöliittymä, jonka avulla opettajat voivat päivittää oman ryhmänsä opetussuunnitelmatiedot tietokantaan, mistä tiedot tulostetaan suoraan koulun verkkosivustolle. Lisäksi tarkoituksena oli lisätä verkkosivustoon mahdollisuus tarkastella yksittäisiä kursseja, niiden sisältöä sekä niin halutessa lisäksi myös yksittäisen lukukauden tai -vuoden kurssisisältöä.

Tähän opinnäytetyöhön valikoitui Qt-ohjelmointiympäristö sen alustariippumattomuuden sekä monipuolisen ohjelmointityökalujensa ansiosta.

Opinnäytetyöni käsittelee Qt-ohjelmointiympäristöä, Linux-käyttöjärjestelmää, niiden historiaa, PHP-ohjelmointia, MySQL-tietokantoja sekä edellä mainittujen avulla toteutettavaa opetussuunnitelmasovellusta.

Lopuksi tarkastelen opinnäytetyöni aiheena olleen opetussuunnitelmasovelluksen kehitysmahdollisuuksia, eli miten tätä kyseistä sovellusta voitaisiin kehittää siten, että sen avulla voitaisiin halutessa laatia ja toteuttaa Kemi-Tornion ammattikorkeakoulun muita tarvitsemia analyysejä, esimerkiksi kompetenssianalyysia tai vastaavaa.

## 2. TAVOITTEET

Opinnäytetyön tavoitteena oli perehtyä Qt-ohjelmointiympäristöön ja sen tarjoaman alustariippumattomuuden avulla suunnitella ja toteuttaa käyttöliittymä, jonka avulla opettajat voisivat itse tallentaa opetussuunnitelmatiedot palvelimella olevaan tietokantaan. Tietokannasta tallennetut tiedot oli tarkoitus ladata suoraan koulun verkkosivustolle.

Tarkoitus oli tutustua kyseiseen ohjelmointiympäristöön ja sen myötä C++ -ohjelmointia apuna käyttäen laajentaa kyseisen sovelluksen käytettävyyttä sellaiseksi, jonka avulla Kemi-Tornion ammattikorkeakoulun opettajat voisivat itse halutessaan päivittää ja ylläpitää tietokannan kautta koulun verkkosivustolla olevia koulutusohjelmakohtaisia opetussuunnitelmatietoja.

Tästä seurauksena oli se, että projektin sisältö laajeni koskemaan myös tietokannan suunnittelun ja käytettävyyden huomioimisen tässä opinnäytetyössä.

Suurimpia haasteita tässä kohtaa oli näin suuren kokonaisuuden hahmottaminen eli se, että projektin osat saataisiin pilkottua niin pieniin osakokonaisuuksiin, että koko projektin läpivieminen onnistuisi ilman suurempia ongelmia.

Lopputuloksena oli kokonaisuudessaan tavoitteidenmukainen kokonaisuus, jota halutesaan voitaisiin laajentaa ominaisuuksiltaan vielä enemmän palvelevammaksi, koulun tarpeita ajatellen.

Tulevaisuuden visiona olisi muovata tästä kyseisestä sovelluksesta sellainen, että sen avulla voitaisiin toteuttaa myös Kemi-Tornion amk:n kompetenssianalyysi sekä mahdollisesti muutama muu osakokonaisuus.

## 3. TYÖVÄLINEET

Tämän opinnäytetyön yhtenä tarkoituksena oli tutustua Qt-ohjelmointiin, minkä vuoksi kyseinen ohjelmointiympäristö, monipuolisine ohjelmointityökaluineen oli luonnollinen valinta. Qt on parhaimmillaan Linux-käyttöjärjestelmässä, mistä johtuen kyseistä käyttöliittymää alettiin suunnitella toteutettavaksi juuri Linux-pohjaisena. Qt sinällään on suoraan käännettävissä myös muille alustoille, joten tämä ohjelmisto olisi käytännössä voitu toteuttaa alustariippumattomuuden ansiosta myös muilla käyttöjärjestelmillä.

### 3.1. QT SDK

Qt SDK on alustariippumaton ohjelmointiympäristö, joka sisältää C++ -luokkakirjaston lisäksi QtCreator-ohjelmointityökalun, jonka avulla varsinkin C++-kielisten ohjelmistojen tekeminen on kohtuullisen helppoa ja vaivatonta. Alla käyn läpi hieman tarkemmin tämän kokonaisuuden historiaa, pääpiirteitä ohjelmistosta ja sen sisältämistä ominaisuuksista.

#### 3.1.1. Qt:n historia

Qt:n kehittänyt Norjalainen Trolltech on perustettu vuonna 1994. Alkuperäistä Qt-rajapintaa alettiin ensimmäisen kerran suunnitella jo vuonna 1992, josta ensimmäinen kaupallinen versio julkaistiin 1994. Tällä hetkellä Qt on käytettävyytensä ansiosta saatavilla useisiin erilaisiin ympäristöihin. /5/

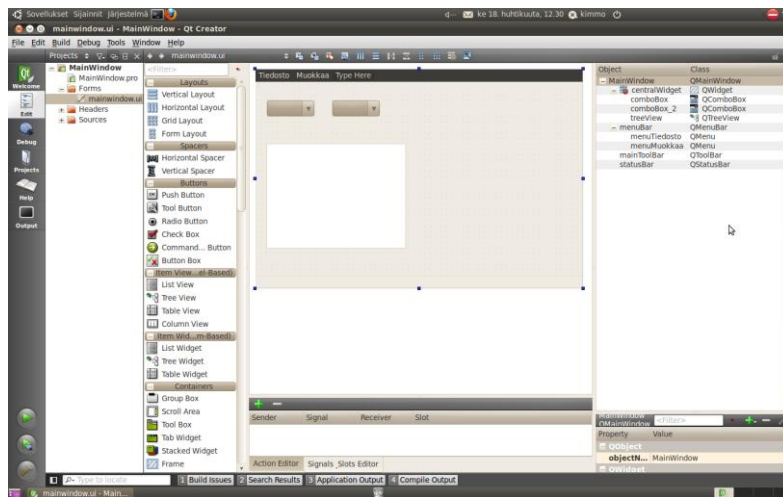
Nokia osti Qt:n vuonna 2008 ja jatkoi sen kehittämistä ahkerasti julkaisten 4.5- sovelluskehityksen maaliskuussa 2009. Qt itsessään sisältää C++ -luokkakirjaston ja alustariippumattoman ohjelmointiympäristön. /5/

Qt:n avulla on kehitetty ohjelmistoja kuten Google Earth, KDE SC, Skype, VirtualBox ja VLC media player. Qt-ympäristöä kehittää Qt Development Frameworks, jonka Nokia omistaa. Qt-ympäristö on vapaa, avoimen lähdekoodin ohjelmisto. /5/



### 3.1.2. Qt Creator

Qt sisältää ohjelmoijalle tärkeitä työkaluja. Näistä yksi tärkeimmistä on Qt Creator, (kuva 1). joka on räätälöity Qt-kehitykseen, mutta jota voidaan käyttää myös yleisesti C++-kehityksessä. /6/



Kuva 1. Qt Creator IDE

Kehitysympäristön toimivuus on yksi tärkeimmistä osista hyvässä ohjelmointikokemuksessa. Qt Creatorissa on todella panostettu sen kehittämiseen. Qt Creatorista löytyy ominaisuuksia, aina koodin täydennyksestä syntaksin tarkastukseen kirjoituksen yhteydessä. Edellä mainitut ominaisuudet, lisättynä nopeuteen, tekevätkin Qt Creatorista kilpailukykyisen muihin avoimen lähdekoodin IDE:ihin verrattuna. /6/

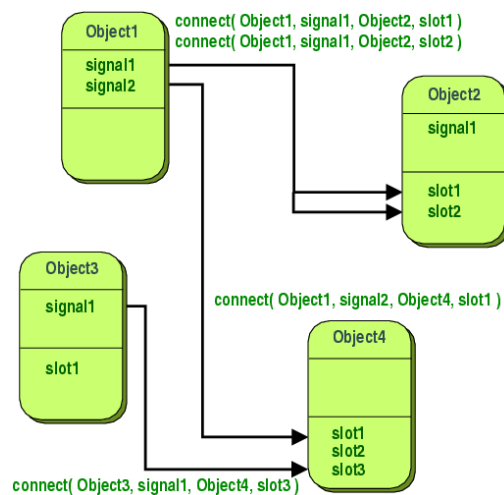
Qt Creator luottaa alustasta löytyviin työkaluihinsa. Toisaalta koska työkalut ovat alustasta riippuvaisia, ei integrointi aina toimi saumattomasti. Käyttökokemus ei ole kuitenkaan aivan samalla tasolla kuin esimerkiksi Visual Studiossa, mutta varsinkin Linuxissa Qt Creator on oikein toimiva työkalu C++ -kehitykseen, vaikka itse Qt ei olisikaan kohdeympäristö. Qt-ohjelmistoa kirjoitettaessa, Qt Creator on oiva valinta kaikilla alustoilla juuri sille räätälöityjen ominaisuuksiensa ansiosta. /6/

Omat haasteensa, kuten aina, tuo natiivikoodin kirjoittaminen. C++ on kielenä monimutkainen. Toisaalta Qt helpottaa myös C++-sovelluskehitystä, tarjoten siihen hyvät työkalut.

QML on esimerkiksi teknologia, jonka käyttöliittymien esittelemä toteuttamismalli tuntuu varsin luontevalta. /6/

Loppupäätelmänä todettakoon, että vaikkakin Qt helpottaa ohjelmistokehityksessä alkuun pääsemistä, parantaen tuottavuutta, on silti syvällisemmästä C++ -osaamisesta hyötyä myös Qt:ta käytettäessä. /6/

### 3.1.3. Signal-slot-mekanismi



**Kuva 2. Signal and slots**

Qt:stä alkunsa saanut ajatus, on tapahtumienkäsittelyyn tarkoitettu Signal-slot-mekanismi, (kuva 2), jonka on Qt:n kehittäjä Eirik Chambe-Eng ideoi. Varsinaisesti Qt:lle tärkeän ominaisuuden perusajatuksena toimii se, että sovelluksessa oleva komponentti lähettää tapahtumaan liittyvän signaalin ja tämä taas kytketään johonkin slotiin. Slotilla tarkoitetaan tässä tapauksessa funktioita, johon signaali on kytketty connect()-funktiolla. tavallisin signal-slot-käyttökohde ovat valikoissa tehdyt klikkaukset sekä erilaiset nappien painallukset ja. /5/

```
Connect (ui->pushbutton_on_triggered, SIGNAL (clicked ()), this, SLOT (calculate ()));
```

## Qt:n lisensioinnit

Qt on Open source-tuote, mutta siitä huolimatta, lisensiointivaihtoehtoja on olemassa useita. Qt 4.5 lisenssejä ovat:

- LGPL, eli *Lesser General Public License*
- GPL, eli *General Public License*
- kaupallinen lisenssi. /5/

## Qt-moduulit

Qt-ohjelmointiympäristöön kuuluvat seuraavat moduulit:

- QtCore
- QtScripts
- QtXml
- QtOpenGL
- QtWebkit
- QtNetwork
- QtSvg
- QtGui
- QSql
- phonon. /5/

## 3.2. PHP

PHP on suosittu nettisivujen toteutukseen tarkoitettu ohjelmointikieli. PHP-koodi suoritetaan palvelimella ennen varsinaisen verkkosivun lähettämistä selaimelle, minkä ansiosta PHP ei vaadi varsinaista tukea selaimelta ja sillä voi käsitellä esimerkiksi palvelimen tiedostoja ja tietokantoja. PHP:n vahvuuksia ovat helppokäyttöisyys ja monipuolisuus, minkä lisäksi se on käytettävissä useimmissa edullisissa webhotelleissa. /7/

### 3.2.1. Historia

*Personal Home page*, eli PHP:n loi aikoinaan Rasmus Lerdorf ja alun perin PHP oli vain joukko makroja, jotka oli tehty helpottamaan koodaajaa sivustonsa ylläpitämisessä.

Kehityksen alussa koko PHP kirjoitettiin uudestaan ja vuonna 1997 sen lomakkeidenkäsittelyominaisuudet houkuttelivat lisää kehittäjiä mukaan. Nykyään se on jo oma ohjelmointikielensä, jolla voidaan toteuttaa todella laajoja tietokantapohjaisia WWW-ympäristöjä. Kun PHP-koodilla rakennetaan sivustoa, on muutama asia jotka on hyvä muistaa. Tiedot tallennetaan aina tiedostot loppupäätteellä .php, jotta palvelimella oleva tulkki tunnistaa koodin PHP:ksi. /7/

PHP on ohjelmointikieli, jota käytetään erityisesti Web-palvelinympäristöissä dynaamisten web-sivujen tekemiseen. Ohjelmointikielen lisäksi PHP-ympäristö sisältää laajan luokkakirjaston. PHP on komentosarjakieli, jossa ohjelmakoodi tulkitaan vasta ohjelman suoritusvaiheessa. /7/

PHP/FI 2.0 julkaistiin marraskuussa 1997 ja sillä oli useita tuhansia käyttäjiä ympäri maailmaa. /7/

PHP:ssä on valmiiksi olemassa eräitä keskeisiä funktioita, joita apuna käyttäen, esimerkiksi tietokantapohjaisten verkkosivustojen operoiminen on todella helppoa. Alla esimerkkejä muutamista tärkeimmistä funktioista:

`mysql_affected_rows` = palauttaa kyselyn muuttamien rivien määrän

`mysql_close` = sulkee yhteyden palvelimelle. (ei pysyvä yhteys)

`mysql_connect` = avaa käyttäjä tunnuksien perusteella yhteyden MySQL-palvelimelle.

`mysql_fetch_array` = hakee kyselyn tuloksen taulukkomuodossa.

`mysql_query` = tekee kyselyn tietokantaan ja palauttaa tuloksen sisäisessä muodossa.

`mysql_select_db` = valitsee tietokannan nimen perusteella. /7/

### 3.2.2. PHP koodi

PHP-kielessä on seuraavat yksinkertaiset tietotyypit (scalar datatypes):

- boolean (totuusarvo)
- integer (kokonaisluku)
- floating-point number eli float (liukuluku)

- string (merkkijono) /7/

Lisäksi kielessä on seuraavat kootut tietotyytit (compound datatypes)

- array (taulukko)
- object (olio) /7/

Kokonaislukuja PHP:ssä voidaan käsitellä kuten C-kielessä. Kokonaisluku voi olla positiivinen tai negatiivinen. Alla eriteltyinä neljä tapaa määrittellä muuttujia, joiden tietosisältönä on kokonaisluku:

`$posluku = 1234; // positiivinen luku`

`$negluku = -123; // negatiivinen luku`

`$oktluku = 0123; // oktaaliluku (83 desimaalilukuna)`

`$hexaluku = 0x12; // heksadesimaaliluku (18 desimaalilukuna) /7/`

Tietokantapohjaisen tulostuksen tekeminen PHP:llä on kohtuullisen yksinkertaista, mikäli omaat hieman perusosaamista molemmista aiheista. Tällaista tietokantapohjaista tulostusta sovellettiin tämän opinnäytetyön verkkosivuston eri vaiheissa monella eri tavalla. Alla esitetty kuva verkkosivustolle muodostuvasta tulostuksesta, kuva 3.

### Tekniikka

Tietotekniikan koulutusohjelman opetussuunnitelma 2008-2012

Koodi	Perusopinnot	1.syk	1.kev	2.syk	2.kev	3.syk	3.kev	4.syk	4.kev
	Tietotekniikan perusopinnot								
3Y2401	Opiskelutaidot ja ammatillinen kasvu		3op						
3T5000	Tietotekniikan perusteet		3op						
3T5090	Tietotekniikanoppimisen projekti		6op						
3Y2402	Tiedonhaku ja e-oppiminen		3op						
	Suomenkieli ja viestintä								
3Y2003	Kirjoittamisen ja esiintymistaidon kurssi				3op				
	Ruotsin kieli ja viestintä								
3Y2304	Ruotsin kieli ja viestintä								
	Englannin kieli ja viestintä								
3Y2301	English communication skills for engineering								
3Y2302	Business english for engineering								
	<b>Ammattiopinnot</b>								
	Sähkötekniikan perusopinnot								
3S0521	Sähköpiirien perusteet								

**Kuva 3. Tietokantatulostus PHP:llä.**

### 3.2.3. Open Source-tuote

PHP on Open Source-tuote, eli sitä voi halutessaan käyttää ja muokata vapaasti niin kaupalliseen, kuin ei-kaupalliseen toimintaan. Tämä tarkoittaa samalla sitä, että hyvin ylläpidetyt Open Source-projektit ovat edullisia myös käyttäjille. Yksi parhaimmista ominaisuuksista vapaanlähdekoodin ohjelmistoissa on se, että mikäli jossain kohtaa johonkin ongelmaan ei heti löydy vastausta, voi tuon ongelman lähettää yhteiselle postituslistalle, jonka tuloksena voi olla selkeä ja älykäs vastaus ongelmaan. /7/

### 3.2.4. PHP siirrettävyys

PHP on kehitetty siten, että sitä voidaan käyttää useissa eri käyttöjärjestelmissä, samalla toimien useiden erilaisten palvelimien sekä tietokantojen kanssa. Tämä tarkoittaa sitä, että voit halutessasi tehdä tuotteen UNIX-ympäristöön ja siirtää sen toiselle alustalle ilman mitään ongelmia. /7/

Voit halutessasi testata tekemäsi tuotteen Personal Web Server-palvelimessa ja asentaa sen UNIX-järjestelmään, jossa PHP on Apachen moduulina. /7/

## 3.3. MYSQL

MySQL on relaatiotietokantojen hallintajärjestelmä, RDBMS (*relational database management system*). MySQL varastoi kaikki tietokannat, taulut, sarakkeet ja rivit sekä käsittelee näitä yhtenä kokonaisuutena. MySQL on laajimmin käytössä oleva, avoimen lähdekoodin tietokanta. Käyttäjiä sillä on miljoonia, yksityisistä ihmisistä aina isoihin yrityksiin. MySQL:llä voidaan prosessoida satoja kyselyjä sekunnissa ja jopa miljoonia päivässä. /4/ MySQL:n parhaita ominaisuuksia ovat sen nopeus, siirrettävyys, hinta sekä yhteensopivuus minkä tahansa ohjelmointikielen kanssa. /7/

### 3.3.1. Historia

MySQL:ää kehittää ruotsalainen yritys MySQL AB, jonka Sun Microsystems osti 16. tammikuuta 2008. MySQL on saatavissa vapaalla GNU GPL-lisenssillä tai kaupallisella, mikäli GPL ei ole sopiva. /7/

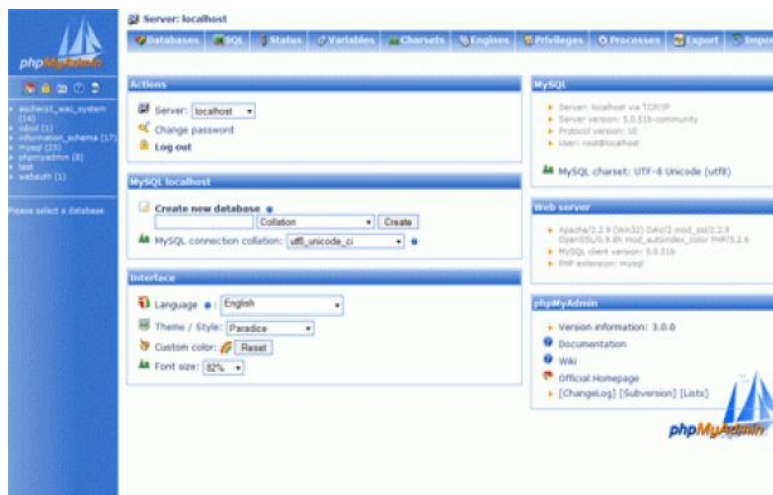
Monista kaupallisista tietokantajärjestelmistä poiketen MySQL:n hallinnointi tapahtuu joko komentoriviltä tai tekstipohjaisella asiakasohjelmalla. /7/

### 3.3.2. phpMyAdmin

MySQL-tietokannan ylläpitoon on olemassa monia keinoja sekä graafisia Web-käyttöliittymiä. Eräs yksinkertaisimmista, samalla monipuolisin vaihtoehtoja on phpMyAdmin Web-käyttöliittymä. Sen avulla tietokannan luominen ja ylläpito on tehokasta ja helppoa. /7/

Tietokantaa voi phpMyAdminilla käyttää paikallisesti tai Internetin yli. Ammattikäyttäjää phpMyAdminin pelkistetty yksinkertaisuus saattaa ärsyttää, mutta guruksi vasta tulossa oleville se on oiva tapa sukeltaa tietokantojen mystiseen maailmaan. /7/

PhpMyAdmin-käyttöliittymä on hyvin selkeä. Kaikki toiminnot, kuten tietokannan, sen sisältämien taulujen ja tietueiden lisääminen sekä poisto tapahtuu graafisessa käyttöliittymässä, joten phpMyAdmin-ohjelman oppiminen on vasta-alkajalle helppoa sekä vaivatonta. Kuva 4. /7/



Kuva 4. PhpMyAdmin

Lisätietoa phpMyAdminista löytyy sovelluksen kotisivuilta. Koska phpMyAdmin toimii selaimella, se on melko käyttöjärjestelmäriippumaton ja toimii yhtä hyvin niin Windowsilla, Macintoshilla kuin Linuxillakin. /7/

## 3.4. LINUX

Linux-käyttöjärjestelmä kehitetään ja levitetään useana eri jakeluna. Jakelun jokainen versio sisältää asennusohjelman, käyttöjärjestelmään kuuluvat ohjelmat sekä ohjeet ja usein suuren määrän muita paketteja. Tässä opinnäytetyössä käytössä oli Ubuntu 10.04 versio. /3/

### 3.4.1. Historia

Linux on Unix-yhteensopiva käyttöjärjestelmä, jossa ydin on ohjelmoitu kokonaan uudestaan. /3/

Unixin varhaisimmat ideat luotiin jo vuonna 1965 MIT:ssä aloitetussa MULTICS-käyttöjärjestelmäprojektissa. Kehitystyössä mukana olleet AT&T:n Bell laboratoriot vetäytyivät projektista 1969, jonka jälkeen Dennis Richie ja Ken Thompson aloittivat sitä yksinkertaisemman moniajokäyttöjärjestelmän ja monen käyttäjän käyttöjärjestelmän kehittämisen. /3/

Nimi Linux tulee Linux-ytimeistä, jonka alun perin kehitti Linus Torvalds vuonna 1991. /2/ Linux sai alkunsa kun Linus Torvalds 1990-luvun alussa kyllästyi MS-DOS käyttöjärjestelmään ja päätti toteuttaa kokonaan uuden käyttöjärjestelmän Intelin edullisille x86-prosessoreille. Tällöin oli olemassa jo Minix, joka oli vapaa miniUnix PC-koneille. Tämä kyseinen käyttöjärjestelmä ei kuitenkaan Linus Torvaldsille riittänyt, vaan hän ohjelmoi tuolloin toimivan esiversion Linux-käyttöjärjestelmästä. /3/

Ensimmäiset rivit Linux-käyttöjärjestelmän historiasta oli kirjoitettu. Tänäpä tuota kyseistä järjestelmää on kehittämässä suuri joukko alan ammattilaisia ja juuri avoimesta lähdekoodista johtuen Linuxia kehitetään koko ajan. /3/



### 3.4.2. Linux-kernel

Nimi Linux viittaa Linux-ydintä käyttävien Unixin kaltaisten käyttöjärjestelmien perheeseen. Linuxia voi käyttää monissa tietokonelaitteissa, muun muassa matkapuhelimissa, taulutietokoneissa, pelikonsoleissa ja palvelimissa. /3/

Monoliittinen ydin eli monoliittinen kernel on kokonainen käyttöjärjestelmä yhdessä osoitevaruudessa ja se tarjoaa tietokoneen osat sekä oheislaitteet erilaisten ohjelmien käyttöön sekä huolehtii käyttöoikeuksista ja resurssien jaosta prosessien sekä eri käyttäjien kesken. Monoliittisen käyttöjärjestelmän, kuten Linuxin tapauksessa käyttöjärjestelmä toimii kokonaan yksin ydintilassa (ring 0) ja suojatussa tilassa (supervisor mode). Monoliittinen käyttöjärjestelmä tarjoaa muille ohjelmistoille rajapinnat laitteiston käyttämiseksi ja hallitsee kaikkia muita ohjelmistoja. Monoliittista käyttöjärjestelmää ei pidä sekoittaa Asiakas-Palvelin (Server-Client) -käyttöjärjestelmäarkkitehtuuriin jossa ydin (mikroydin) on vain osa käyttöjärjestelmästä. /3/

Monoliittinen arkkitehtuuri eroaa muista käyttöjärjestelmäarkkitehtuureista merkittävästi. Se on alkuperäinen 60-luvulla kehitetty käyttöjärjestelmäarkkitehtuuri, jolla ensimmäiset Unixit suunniteltiin. Tuolloin ohjelmisto, joka tunnetaan nykyisin käyttöjärjestelmänä, tunnettiin paremmin muilla nimillä kuten *Supervisor*, *Master Program*, *Kernel*, *Core* ja *Operating System*. Myöhemmin monoliittinen rakenne alkaa tuottaa ongelmia kun käyttöjärjestelmän koko kasvoi miljooniin riveihin eivätkä mahtuneet enää tietokoneen keskusmuistiin. Pahimpia ongelmia olivat kuitenkin bugit. Käyttöjärjestelmän toimiessa yhdessä osoitevaruudessa ongelmia tuotti se, ettei käyttöjärjestelmän tietyn osan virheitä voitu eristää muista osista. Esimerkiksi virhettä verkkoprotokollassa ei voitu eristää vaan se vaikutti myös kaikkiin muihin käyttöjärjestelmän toimintoihin ja kaatoi käyttöjärjestelmän, jolloin koko ohjelmistojärjestelmä kaatui. Käyttäjä sai vain virheilmoituksen käyttöjärjestelmän kaatumisesta ruudulle, jota kutsuttiin tuolloinkin Kernel Panic:ksi. Myös lähdekoodin ylläpidettävyys muodostui hyvin hankalaksi, jos ei mahdolliseksi. Hyvin harva pysyi selvillä käyttöjärjestelmän toiminnoista mikä hidasti ja vaikeutti kehitystyötä vielä lisää. /3/

Rinnalle kehittyi uudenlainen käyttöjärjestelmän arkkitehtuuri, niin kutsuttu Palvelin-Asiakas (Server-Client). Palvelin-Asiakas -arkkitehtuurissa monoliittinen käyttöjärjestelmä jaoteltiin moduuleiksi joita kutsuttiin palvelimiksi. Jokainen palvelin suoritti käyttöjärjestelmän eri tehtävät itsenäisesti ja tarjosi palvelunsa niitä tarvitseville prosesseille. Jokaista palvelinta hallitsi mikroydin, hyvin yksinkertaiseksi suunniteltu ydin joka toteutti vain tärkeimmät käyttöjärjestelmän tehtävät. Jokainen palvelin toimi suojattuina prosesseina mikroytimeen ja ne olivat suojattu vielä jokaiselta toiselta palvelimelta, jolloin yhden palvelimen kaatuminen ei teoriassa vaikuttanut muihin mikroytimeen ja muihin palvelimiin /3/

### 3.4.3. Komentotulkki

Erilaisissa Unix- ja Linux-järjestelmissä on käytössä lukuisia erilaisia komentotulkkia. Yleisimmin tavallisessa Linux-asennuksessa tulee mukana 3-4 erilaista komentotulkkiä, joista jokainen löytää mieleisensä. Tämän hetkisisissä Linux-jakeluissa oletusarvoisena komentotulkkinä on bash. Oman jakelunsa komentotulkin saa helpoiten selville komennolla `echo $SHELL`. /3/

Alla on lueteltuna erilaisia komentokehotteita ja niiden sisältämiä ominaisuuksia:

*sh*

Alkuperäinen Bourne Shell. Ensimmäinen Linux-komentotulkki 70-luvulta, josta puuttuu yleisimpiä ominaisuuksia siinä määrin, ettei tätä kyseistä tulkkiä enää nykypäivänä kukaan juuri käytä. /3/

*bash*

Bourne Again Shell on uudistettu versio vanhasta Bourne Shellistä. Sisältää paljon kehittyneitä ominaisuuksia, esimerkiksi vanhasta Bourne Shellistä puuttuvan komentorivin editoinnin sekä komentohistorian. Uudempi Bourne Again Shell on yhteensopivan vanhan version kanssa. Kyseinen komentotulkki on yleisimmin käytetty Linux-komentotulkki. /3/

*csh*

C shell on komentorivillä käytettynä hyvin samankaltainen kuin Bourne Shell, mutta sen ohjelmointiominaisuudet ovat hyvin erilaisia. C shell on myös vanhanaikainen ja sen vuoksi vähän käytetty komentotulkki. /3/

## 4. OHJELMISTON SUUNNITTELU

Ohjelmiston suunnittelussa käytettiin Microsoft Visio- , sekä NetBeans IDE-ohjelmia, joiden avulla tietokannan, käyttöliittymän sekä näiden sisältämien, eri toimintojen suunnittelu oli varsin helppo toteuttaa.

Käyttöliittymän eri toimintojen suunnittelu tässä opinnäytetyössä toteutettiin laatimalla ensin käyttötapauskaavio, jossa määritellään sovellukselle asetettuja tavoitteita tapauskohtaisesti. Tämän jälkeen laadittiin UML-kaavio ja tällä tavoin pienemmissä osakokonaisuuksissa tarkasteltuna, tarkennetaan vaatimusmäärittely koskemaan mahdollisimman suurta osaa käyttöliittymälle asetetuista tavoitteista. Tämä ehkäisee turhaa työtä, joka oikeassa työelämässä tulisi kovin kalliiksi. Toisaalta tällä varmistetaan osaltaan se, ettei käyttöliittymän varsinaisessa koodaamisvaiheessa, tarvitse enää palata takaisin määrittelyn tarkasteluun, vaan projekti etenee vaiheittain, suunnitellusti eteenpäin.

### 4.1. Suunnittelumenetelmät

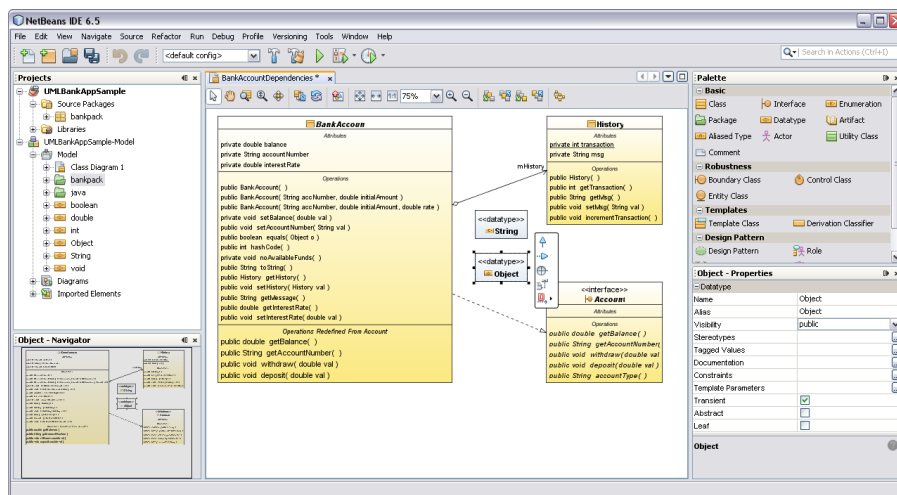
UML on kieli, jolla voidaan kuvata ohjelmistoa suorittavaa järjestelmää, ohjelman toimintaa ja sen sisältöjä. /2/

UML:n kehitys aloitettiin 1994, jolloin omilla tahoillaan oliomallinnustekniikoita kehittäneet Grady Booch ja James Rumbaugh aloittivat työn yhdistääkseen voimansa. Ivar Jacobson tuli mukaan vuonna 1995, tuoden mukanaan oman menetelmänsä. Näiden oliomallinnustekniikoiden yhdistäminen yhdeksi, oli järkevää kolmesta syystä. Menetelmät olivat jo valmiiksi kehittymässä samaan suuntaan, käsitteiden yhdenmukaistaminen toisi vakautta, helpottaen samalla ohjelmistoprojektien yhteensovittamista. Lisäksi menetelmien yhdessä kehittäminen toi tullessaan ratkaisuja ongelmiin, joita ei aikaisemmin kyetty ratkaisemaan yksin. Vuonna 1996 OMG (Object Management Group) julkaisi ehdotuksen UML 1.0 versioksi, minkä pohjalta UML on kehittynyt lukuisten yhteisöjen panostuksen tukemana. UML:n versioista tällä hetkellä 2.0:n ominaisuudet ovat saaneet riittävän työkaluohjelmistojen tuen. /2/

UML on yleiskäyttöinen kieli, tarkoittaen sitä, että sitä voidaan käyttää kaikilla olemassa olevilla toteutusympäristöillä ja sovellusalueilla. Toteutusympäristöllä tarkoitetaan alustan sekä ohjelmointikielen yhdistelmää, kuten J2EE tai .NET. Sovellusalueilla tarkoitetaan sitä olosuhdeympäristöä, jossa ohjelmistoa käytetään kuten terveydenhoito, talous, tietoliikenne. UML-kaavioita käsiteltäessä MOF (*Meta-Object Facility*) eli UML-kielen metamalli toimii säännöstönä kaavioiden piirtämiselle. /2/

XML:ään perustuva XMI (*XML Metadata Interchange*) on formaatti UML-kielen metadatan siirtämiseen. UML on standardin kaltainen tyyli toteutettavissa olevan järjestelmän kuvaukseen oliopohjaisessa ohjelmistotuotannossa. Yleiskäyttöisyys sekä laaja hyväksyntä pitävät UML:n toistaiseksi suosittuna, joskaan ei ainoana soveltuvana tapana ohjelmistojen visualisoinnissa. /2/

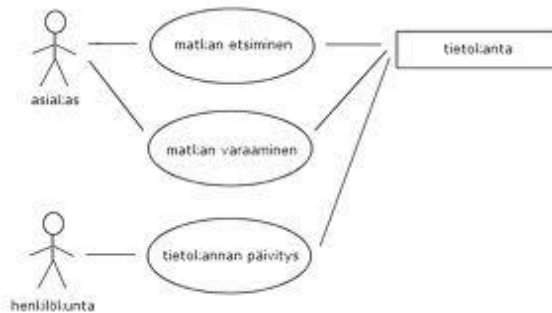
UML-mallinnukseen on olemassa monta käyttökelpoista sovellusta, tässä opinnäytetyössä käytettiin NetBeans IDE:n, kuva 5, UML-mallinnusohjelmaa, joka on kaikkiaan todella korkealaatuinen ohjelmisto UML-mallinnusta ajatellen.



**Kuva 5. NetBeans IDE UML**

Tässä vaiheessa mietittiin siis sitä, mitä toiminnallisuuksia käyttöliittymässä tulisi olla ja miten eri toiminnallisuudet sekä niiden yhteydet toisiinsa toteutettaisiin mahdollisimman yksinkertaisesti ja nopeasti.

Seuraavana suunnitteluvaiheena seurasi UML-kaavion tekeminen. Mallinnuksessa tarkoituksena on määritellä käyttöliittymän funktiot, niiden yhteydet sekä toiminnallisuus. Tähän suunnitteluun sopi parhaiten nk. käyttötapauskaavion luominen. Käyttötapauskaaviossa, kuva 6, määritellään sovelluksen käyttäjät, käyttäjien toimet, sovelluksen tapahtumat sekä se kuinka kyseiset asiat ovat yhteydessä toisiinsa.



**Kuva 6. Käyttötapauskaavio**

Käyttötapauskaavio tukee käyttötapojen suhteiden kuvaamista, ei niiden sisällön kuvaamista. UML ei määrittele sisällön esitystapaa.

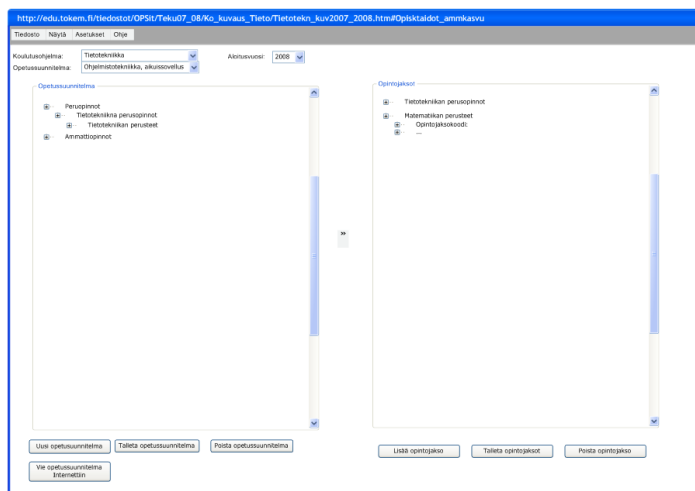
- Käyttötapausten tulee olla ymmärrettävissä sekä asiakkaan että suunnittelijan kannalta.
- Käyttötapausten abstraktiotason on oltava sopiva (ei esim. yleensä käyttöliittymän yksityiskohtia).
- Kaikkia käyttötilanteita/asiakasvaatimuksia ei voi/kannata antaa käyttötapauksina.
- Käyttötapausten tulisi olla suhteellisen lyhyt.
- Käyttötapaus tuottaa käyttäjälle lisäarvoa, joskaan ei yleensä yksittäinen ohjelmiston toiminto, vaan kokonainen tuloksen tuottava ketju toimintoja.

- Käyttötapaus ei siis yleensä ole yksittäinen ohjelmalla suoritettava toiminto, eli ei siis esimerkiksi: tekstin kopiointi leikkuupöydälle. /1/

## 4.2. Käyttöliittymän suunnittelu

Varsinainen käyttöliittymän suunnittelu aloitettiin sillä, että mietittiin, miten aikaisemmissa määrittelyissä laaditut toiminnallisuudet tultaisiin varsinaisessa käyttöliittymässä toteuttamaan mahdollisimman selkeästi ja samalla yksinkertaisesti. Tässä kohtaa hahmoteltiin ensimmäinen käyttöliittymäluonnos Microsoft Visiota apuna käyttäen. Alla olevassa kuvassa on esitelty käyttöliittymän lopullinen luonnos.

Kuten alla olevasta hahmotelmasta (kuva 7), käy ilmi, päätimme toteuttaa käyttöliittymän MainWindow-tyyppisenä, jolloin käyttöliittymään on mahdollisuus luoda haluamansa valikot, joiden avulla itse toimintojen valitseminen on helppoa ja sujuvaa.



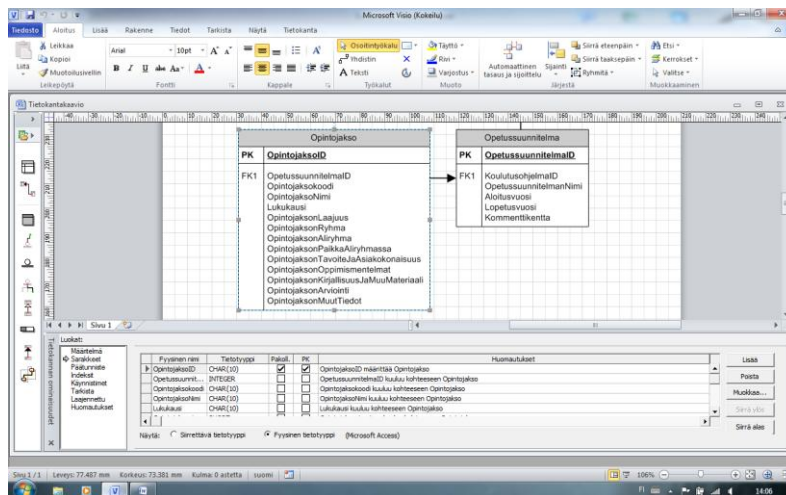
**Kuva 7. Käyttöliittymäluonnos.**

Tämän jälkeen aloimme miettiä sitä, mitä toiminnallisuuksia käyttöliittymässä tulisi olla ja miten eri toiminnallisuudet sekä niiden yhteydet toisiinsa toteutettaisiin mahdollisimman yksinkertaisesti ja nopeasti.

### 4.3. Tietokannan suunnittelu

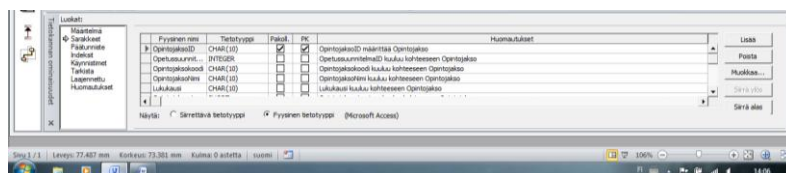
Tietokannan suunnittelu aloitettiin sillä, että mietittiin, kuinka monta varsinaista tietokantataulua tarvitsemme sekä mitä tietueita noihin tauluihin tulisi sisällyttää. Lisäksi pohdimme sitä, minkälaisia yhteyksiä kyseisessä tietokannassa toivottiin olevan ja miten vaaditut yhteydet tultaisiin toteuttamaan.

Tässä kohtaa hahmoteltiin ensimmäinen tietokantaluonnos Visiota apuna käyttäen, kuva 8. Alla olevassa kuvassa on esitelty tietokannan taulukkorakenne sekä niiden väliset yhteydet.



Kuva 8. Tietokantasuunnitelma

Microsoft Visiossa on sellainen erinomainen ominaisuus, että suunnitteluvaiheessa voidaan valmiiksi määrittellä tietokantatietueiden ominaisuudet sekä määrittelykset, ja tällä tavoin helpotetaan varsinaista koodaustyötä. Kuva 9.



Kuva 9. Tietueiden arvot.



#### 4.4. Verkkosivuston suunnittelu

Verkkosivuston suunnittelu aloitettiin sillä, että mietittiin, minkälaisia toimintoja sivustolla tulisi olla kyseistä sovellusta ja tietokantaa yhteen sovitettaessa. Tässä vaiheessa päädyttiin siihen, että sivusto toteutettaisiin eräänlaisilla table-elementeillä, joiden sisälle linkitykset sekä tarvittava tieto sitten tietokannasta tulostettiin. Alla olevassa kuvassa on esitelty verkkosivustolle tulostettuja tietoja, jotka on suoraan tietokannasta haettu. Kuva 10.

##### Tekniikka

###### Tietotekniikan koulutusohjelman opetussuunnitelma 2008-2012

Koodi	Perusopinnot	1.syk	1.kev	2.syk	2.kev	3.syk	3.kev	4.syk	4.kev
	Tietotekniikan perusopinnot								
3Y2401	Opiskelutaidot ja ammatillinen kasvu	3op							
3T5000	Tietotekniikan perusteet	3op							
3T5090	Tietotekniikanoppimisen projekti	6op							
3Y2402	Tiedonhaku ja e-oppiminen		3op						
	Suomenkieli ja viestintä								
3Y2003	Kirjoittamisen ja esiintymistaidon kurssi				3op				
	Ruotsin kieli ja viestintä								
3Y2304	Ruotsin kieli ja viestintä								
	Englannin kieli ja viestintä								
3Y2301	English communication skills for engineering								
3Y2302	Business english for engineering								
	<b>Ammattiopinnot</b>								
	Sähkötekniikan perusopinnot								
3S0521	Sähköpiirien perusteet								

#### Kuva 10. Verkkosivustosuunnitelma

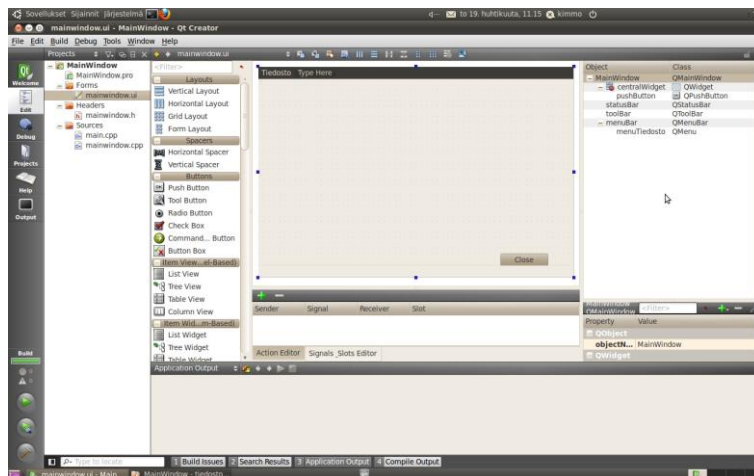
Seuraavassa vaiheessa mietittiin sitä, kuinka helposti sekä varsinkin selvästi saataisiin tulostettua tiedot varsinaisista opetussuunnitelman opintojaksoista siten, että loppukäyttäjän olisi mahdollisimman helppoa ja nopeaa löytää haluttu tieto. Tämä päädyttiin toteuttamaan siten, että varsinaisiin opintojaksokokonaisuuksiin tulostus vaiheessa lisätään linkki, josta haluttu sisältö tulostuu verkkosivustolle. Tällä tavoin kyseinen tieto löytyy mahdollisimman vaivattomasti.

## 5. OHJELMISTON TOTEUTUS

Ohjelmiston toteutus tässä opinnäytetyössä jakautuu kolmeen eri osakokonaisuuteen: Käyttöliittymään, tietokantakokonaisuuteen sekä nämä kaksi yhteen sitovaan verkkosivustokokonaisuuteen. Seuraavassa kuvailen edellä mainittujen kokonaisuuksien eri vaiheita hieman tarkemmin.

### 5.1. Käyttöliittymän toteutus

Johtuen kokemattomuudestani Qt-ohjelmointiympäristöstä, varsinainen käyttöliittymän toteuttaminen alkoi siitä, että tutustuin Qt Creatorin tarjoamiin ohjelmointityökaluihin hieman tarkemmin, tehden sovelluksen, joka on toiminnoiltaan hyvin yksinkertainen. Uutta projektia luotaessa ensimmäisenä tehtävänä on määrittää se, minkälaisen käyttöliittymän haluaa tehdä ja mitä eri kirjastoja kyseisessä sovelluksessa halutaan käyttää. Tämän jälkeen päästään varsinaisten ominaisuuksien toteuttamiseen, mikä tässä opinnäytetyössä oli kohtuullisen helppo toteuttaa, johtuen hyvin valmistellusta ohjelmistosuunnittelusta.



**Kuva 11. Ensimmäinen sovellus**

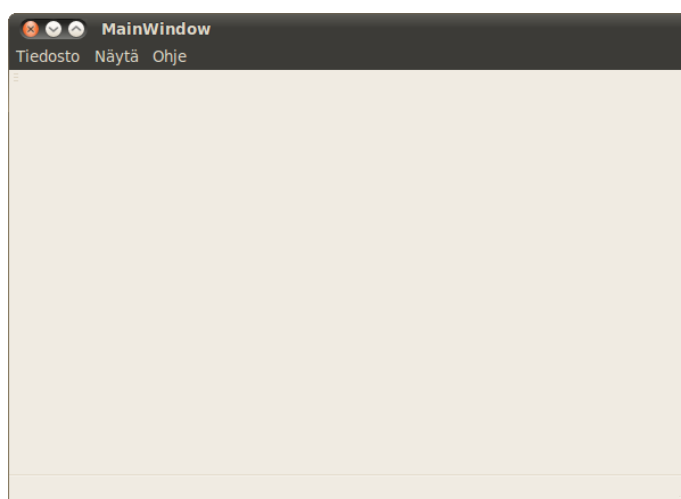
Yllä kuvattuna ensimmäinen sovellus, kuva 11, joka tässä tapauksessa tekee pelkän ikkunan ja siihen painikkeen. Painikkeen toiminnon tekeminen tapahtuu siten, että Qt Creatorissa valitaan kyseinen painike, sen jälkeen valitaan hiiren oikea painike ja avautuvasta valikosta teksti ”go to slot”. Tämän jälkeen editori avautuu mainwindow.cpp-tiedostoon,

johon kyseisen funktion määritelmä tulostuu. Tähän funktion sisälle voidaan sitten kirjoittaa haluttu toteutus, tässä tapauksessa kun painiketta painetaan, sovellus sulkeutuu, joten funktioksi riittää pelkkä `close()`.

Uutta projektia luotaessa on hyvä varmistaa, että jo alkuvaiheessa tulee valinneeksi sovelluksen vaatimat kirjastot mukaan, tässä tapauksessa yhtenä tärkeimmistä, SQL-tietokanta kirjasto, jonka avulla yhteys tietokantaa muodostetaan.

Itse käyttöliittymä on Qt Creatorin avulla todella helppo toteuttaa, eikä käyttöliittymän ulkoasun rakentamisessa loppuen lopuksi mene kovin kauaa. Varsinainen työ alkaa vasta, kun käyttöliittymän eri toiminnallisuuksiin alkaa tehdä haluttuja muutoksia, toisin sanoen funktioita, joista ohjelma ajettaessa tietää, mitä milloinkin on tarkoitus tehdä.

Ensimmäisessä vaiheessa tehtiin sovelluksen Menu-valikot sekä toteutettiin niihin halutut toiminnallisuudet, kuva 12. Valikoiden teossa jälleen huomattiin se, kuinka monipuoliset työkalut Qt Creator sisältää, joten valikoiden teko oli helppoa ja nopeaa. Ensin vain valitaan MainWindow yläreunasta valikko, laaditaan siihen sopiva teksti, jonka jälkeen siirrytään Signal and slot-osioon ja toteutetaan tähän haluttu toiminto. Käyttöliittymän valikoiden kautta tässä sovelluksessa voidaan luoda uusia toimialoja, koulutusohjelmia, opetus suunnitelman eri osia sekä päivittämään haluttuja tietoja halutuilla opintojaksoille.



**Kuva 12. Käyttöliittymä**

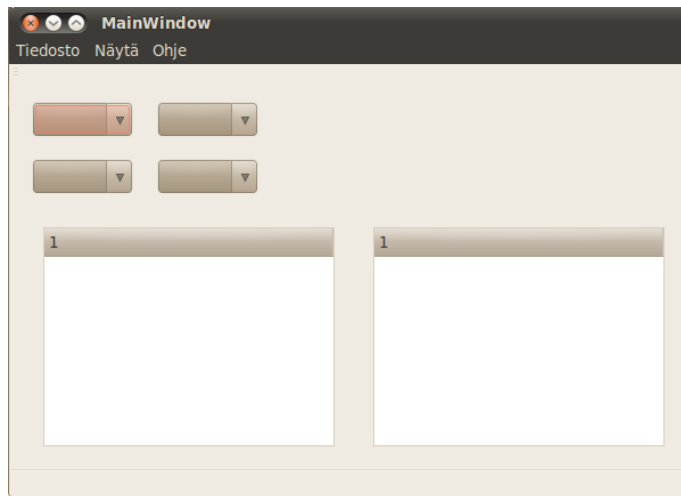
Tässä projektissa helpotusta toi se, että suunnitelma oli tehty valmiiksi niin selväksi, että varsinaisten työvaiheiden pilkkominen oli tapahtunut jo suunnitteluvaiheessa ja tästä johdun eri osakokonaisuuksien työstäminen oli siinä mielessä helpompaa. Yksi osa kerrallaan rakennettaessa, välttyttiin turhalta työltä ja vaikka kyseessä oli todella laaja kokonaisuus, sen pilkkominen pienempiin osakokonaisuuksiin edesauttoi koko projektin toteuttamista.

Valikoiden tekemisen jälkeen käyttöliittymään lisättiin alasvetovalikot, joista voitaisiin valita haluttu toimiala, koulutusohjelma, aloitusvuosi sekä suuntautumisvaihtoehto. Nämä niinkutsutut DropBox:it oli jälleen helppoa lisätä Qt Creatorin avulla. Kuva 13.



**Kuva 13. Drop box**

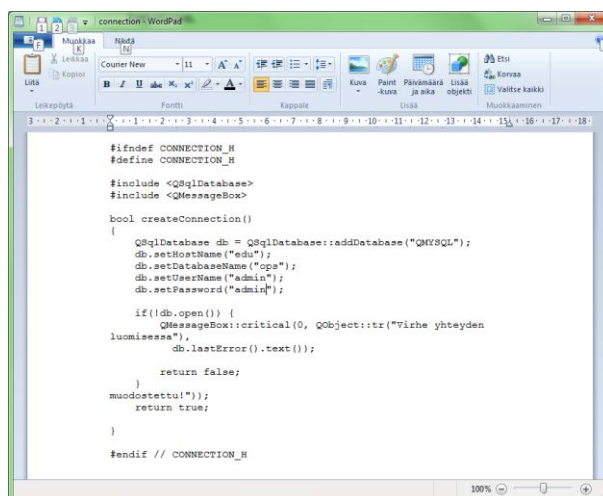
Varsinaisten toiminnallisuuksien toteuttaminen vaati sitten jälleen omien funktioiden toteuttamisen. Tässä kohtaa mietittiin jo sitä, että suunnitteluvaiheessa tietojen lisääminen, poisto ja päivittäminen haluttiin tapahtuvan niin sanotulla drag and drop-menetelmällä, joten siitä syystä käyttöliittymään lisättiin kaksi kappaletta TreeWidget-tyyppisiä valikoita, joihin tieto lisättäisiin ja joista tuo haluttu tieto tallennettaisiin suoraan tietokantaan.



**Kuva 14. Valmis ulkoasu**

Kun itse käyttöliittymä alkoi olla ulkoasultaan valmis, kuva 14, päästiin varsin työlääseen vaiheeseen, eli eri toiminnallisuuden toteuttamiseen. Tämä vaati jokaiselle toiminnolle oman funktionsa ja sen vuoksi seuravana alettiin tutkia Qt:n API-dokumentaatioita, etsien sieltä aina kulloinkin tarpeelliseksi koettu funktio, joka sitten muokattiin halutulla tavalla toimivaksi.

Tietokantayhteyden toimiessa, tarvitaan koodi, jonka avulla ohjelma luo yhteyden tietokantaan. Koodissa määritellään ensiksi tietokanta-ajuri, jota käytetään, sen jälkeen haluttu tietokanta, käyttäjätunnus ja salasana. Alla tarvittavat tiedot connection.h-tiedostossa. Kuva 15.



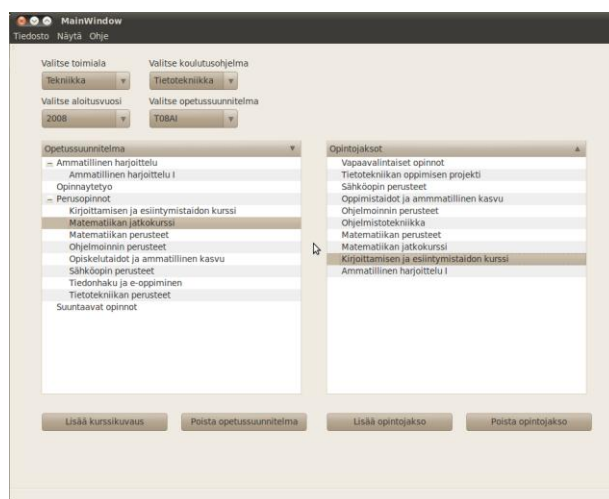
**Kuva 15. Tietokantayhteys**

Seuraavana tarvitsi asentaa tässä opinnäytetyössä tarvittavan MySQL-tietokannan vaatimat ajurit kyseiseen ohjelmointiympäristöön, joka tässä tapauksessa osoittautui luultua haastavammaksi toteuttaa.

Etsin Internetistä tarvittavien ajureiden asennusohjeen, joka sinällään ei ole kovin monimutkainen, mutta tässä tapauksessa asennuksen yhteydessä tähän hetkeen tuntemattomasta syystä kyseiset ajurit sekä niiden tarvitsemat kirjastot asentuivat väärään kansioon, joten suurin työ oli siinä, että ilmenneistä virheilmoituksista ensin sai pääteltyä sen, mitä ohjelman suorituksessa oli vialla ja kuinka tuon kyseisen vian onnistui korjaamaan. Lopulta yhteys saatiin tietokantaan.

Yhteyden luomisen jälkeen voitiin alkaa tehdä lopullisia toimintoja, joiden kanssa meni sitten todella kauan aikaa. Eniten työtä oli siinä, kun tietokannasta haluttiin ensin saada käyttöliittymän valikoihin tarkkaan määritelty tieto, jonka perustella valittiin tieto joka tulostuisi seuraavaan valikkoon ja sen myötä saataisiin varmistettua se, että uusia opetussuunnitelmia luotaessa, haluttu tieto saataisiin jälleen tallennetuksi oikeaan paikkaan tietokannassa.

Viimeisenä tehtävänä oli toteuttaa TreeWidget-valikoihin, hiiren oikealla painikkeella valittavat valikot ja toteuttaa niiden toiminnot halutulla tavalla toteutuviksi. Alla kuvattuna lopullinen käyttöliittymä. Kuva 16.

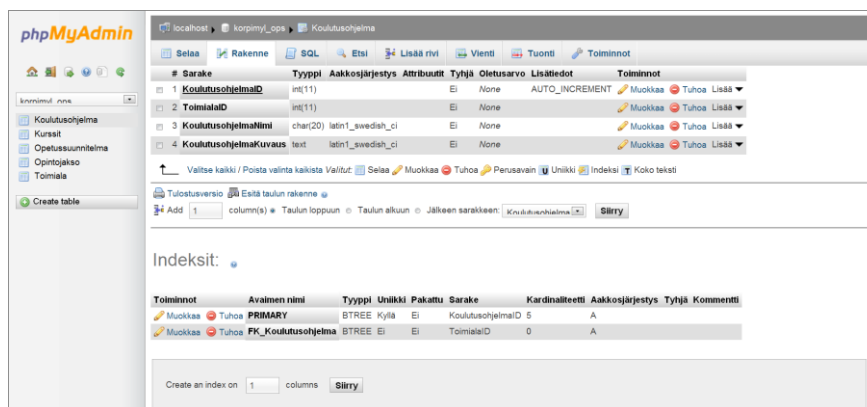


**Kuva 16. Valmis käyttöliittymä.**

## 5.2. Tietokannan toteutus

Tietokannan toteuttaminen tapahtui phpMyAdmin-ohjelman avulla. Valmiin tietokantasuunnitelman pohjalta toteutettuna, tietokannan, taulujen sekä tietueiden tekeminen oli todella helppoa. Aikaa vievää oli käsin kirjoittaen luoda nuo tietueet sekä niiden väliset yhteydet siten, että käyttöliittymästä tallennettu tieto onnistuttiin lataamaan verkkosivustolle ongelmitta.

Taulujen luominen tietokantaan tapahtui siten, että phpMyAdmin-ohjelmassa luodaan uusi tietokanta. CREATE DATABASE mydatabasename-komennolla, jossa mydatabasename on tietokannalle määritelty nimi. Kuva 17.



Kuva 17. phpMyAdmin.

Tämän jälkeen phpMyAdmin-käyttöliittymä pääsee oikeuksiinsa, kun varsinaisten taulujen sekä niiden sisältämien tietueiden luonti tapahtuu suoraan valmiissa käyttöliittymäympäristössä. Taulun lisääminen sekä sen sisältämien kenttien, viiteavainten sekä keskinäisten yhteyksien määrittely onnistuu kohtuullisen vaivatta.

Taulujen luominen tietokantaan tapahtuu komennolla CREATE TABLE. Komennon perään määritellään haluttu määrä tietueita, niiden sisältämä tietotyyppi, merkkimäärä ja loppuksi valitaan aina kyseessä olevan taulun ns. Viite-avain, jonka perustella taulujen sisältämä tieto tulee yksilöidyksi. Alla esimerkkikoodi taulun sekä sen sisältämän tietueen sisällön tuottamisesta.

```
CREATE TABLE `1151_ops`.`Toimiala`
```

```
(`ToimialaID` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,  
`ToimialaNimi` CHAR( 20 ) NOT NULL ,  
`ToimialaKuvaus` TEXT NOT NULL )  
ENGINE = MYISAM ;
```

Määriteltäessä tietokantataulujen välisiä yhteyksiä, on tärkeä muistaa se, että valitut tiedut ovat molemmissa tauluissa, eikä sen vuoksi pääse tapahtumaan mitään epäselvyyksiä. Alla koodiesimerkki edellä mainitun yhteyden luomiseksi:

```
ALTER TABLE `Koulutusohjelma`  
ADD CONSTRAINT FK_Koulutusohjelma  
FOREIGN KEY (ToimialaID) REFERENCES Toimiala(ToimialaID)  
ON UPDATE CASCADE  
ON DELETE CASCADE;
```

Kun vaadittavat taulut sekä niiden sisältämä tieto on tallennettu tietokantaan, alkaa varsinainen käyttöliittymän ja tietokannan yhteensovittaminen.

Tietokannasta on tässä opinnäytetyössä tarkoitus ensin luoda ja tallentaa uusia Toimiala-, Koulutusohjelma-, Opetussuunnitelma-, Opintojakso- sekä kurssikokonaisuuksia, joita sitten tuolta kannasta ladataan käyttöliittymässä sijaitseviin pudotusvalikoihin ennalta käyttöliittymässä määritellyin ehdoin.

Tämän jälkeen pudotusvalikoista valitun tiedon perusteella voidaan käyttöliittymässä oleviin TreeWidget-objekteihin luoda uusia, lisätä, muokata tai poistaa haluttuja kokonaisuuksia, täten luoden toimiva opetussuunnitelma sisältö halutulle koulutusohjelmalle. Tarkoituksena tällä se, että kaikki tieto tallentuu tietokantaan reaaliaikaisesti, sitä mukaa kun tietoa lisätään, poistetaan tai muokataan käyttöliittymässä.

Vaativinta tässä kohdassa on se, että tietokannan rakenne saadaan vastaamaan käyttöliittymässä asetettuja määritelmiä ja sitä kautta välttämään ongelmilta. Tietokannan ja käyttöliittymän jatkuva testaaminen oli yksi tärkeimpiä osa-alueita koko opinnäytetyön ajan.



### 5.3. Verkkosivuston toteutus

Verkkosivustokokonaisuuden toteuttaminen tässä opinnäytetyössä oli näistä kolmesta osakokonaisuudesta laajin, mutta toisaalta yksi vähiten aikaa vieneistä, johtuen siitä että minulla oli kaikista eniten kokemusta PHP-ohjelmoinnista. Toisaalta tässä kohtaa voidaan todeta myös se, että jälleen kerran hyvin suunniteltu oli puoleksi tehty, koska verkkosivustosuunnitelman pohjalta itse toteutus oli kohtuullisen helppoa.

Aluksi rakennettiin sivuston ulkoasu, valikot sekä tulostukseen tulevat taulukot. Päätettiin fontit, sivun värit ja muut ulkoasuun vaikuttavat asiat. Tässä kohtaa apuna toimi jo käytössä olevat edu tokem:in sivut, joista värimaalima saatiin suoraan tähän sivustokokonaisuuteen.

Ensimmäisenä toteutetut linkitykset toteutettiin JavaScript-koodilla, jossa linkkien tieto, haettiin jälleen suoraan tietokannasta. Linkitykset toimivat siten, että käyttäjän valitessa tietty toimiala, sen jälkeen haettiin tämän perusteella sopivat koulutusohjelmat linkkitulosteeseen ja tämän jälkeen koulutusohjelman perusteella valmistumisvuosi ja lopuksi sen perusteella toteutettiin lopullinen tietojen tulostus.

Yhteyden muodostaminen verkkosivustolta tietokantaan tapahtuu seuraavalla koodilla:

```
<?php
$host="localhost";
$user="admin";
$pass="admin";
$db="ops";

$link= mysql_connect($host, $user,$pass);
if(!$link) die("Yhteyttä tietokantapalvelimeen ei saatu");
mysql_select_db($db)or die("Yhteyttä ei saatu tietokantaan $db". mysql_error());
?>
```

Php:ssä on tällaista tietokantahakua varten olemassa varsin laaja valikoima erilaisia funktioita, joista sitten jokainen voi niistä muokata omaan tarkoitukseensa sopivia. Alla kuvattuna linkitysten hakufunktiot koodina:

```
$query = "SELECT * FROM Toimiala WHERE ToimialaNimi='$Toimiala'";
$result =mysql_query($query);
while ($newArray = mysql_fetch_array($result))
{
    $item_Toimiala_ID = $newArray['ToimialaID'];
    $item_Toimiala_nimi = $newArray['ToimialaNimi'];

    print " item_Toimiala_nimi<br>";
}
```

Tämän jälkeen seuraavana toteutuksessa keskityttiin siihen, miten halutut tiedot saadaan ladattua palvelimelta selkeästi taulukoituna, samalla huomioiden sen, että tarvittaessa noissa tulosteissa tuli olla linkit haluttuihin tietoihin, kuten opintojaksojen kuvauksiin.

Nämä tulostukset toteutettiin osaksi HTML-kielen avulla, jossa on valmiina olevia funktioita, joilla erilaisten taulukoiden sekä tietueiden tulostaminen onnistuu ongelmitta. Alla kuvattuna tarvittava koodi tähän tarkoitukseen:

```
$query = "SELECT * FROM Opintojakso JOIN `Opetussuunnitelma` ON Opetussuunnitelma.OpetussuunnitelmaID=Opintojakso.OpetussuunnitelmaID
WHERE      Opetussuunnitelma.Aloitusvuosi='$Aloitusvuosi' AND Opintojakso.Opintojaksoryhma='Perusopinnot' AND Opintojakso.Lukukausi='1.syksy' AND Opintojakso.OpintojaksoPaikkaAliRyhmassa='1'";

$result =mysql_query($query);
while ($newArray = mysql_fetch_array($result))
{
    $item_Opintojakso_ID = $newArray['OpintojaksoID'];
    $item_Opintojakso_koodi = $newArray['Opintojaksokoodi'];
```

```

$item_Opintojakso_nimi = $newArray['OpintojaksoNimi'];
$item_Opintojakso_laajuus = $newArray['OpintojaksoLaajuus'];
$item_Lukukausi = $newArray['Lukukausi'];

print"<tr><td>$item_Opintojakso_koodi</td><td>
<A href=\"kurssikuvaus.php?OpintojaksoNimi=$item_Opintojakso_nimi
&Toimiala=$Toimiala&KoulutusohjelmaNimi=$Koulutusohjelma
\"target=\"omakehys\">$item_Opintojakso_nimi</A></td></tr>
}

```

Alla kuvattuna palvelimella valmiina oleva tulostus taulukoissa. Kuva 18. Teksteissä on linkitykset opinjaksokuvauxsiin, joita käyttäjä voi halutessaan tarkastelle joko yksittäin, lukukausi- tai lukuvuosi-kohtaisesti.

**Tekniikka**

Tietotekniikan koulutusohjelman opetussuunnitelma 2008-2012

Koodi	Perusopinnot	1.syk	1.kev	2.syk	2.kev	3.syk	3.kev	4.syk	4.kev
	Tietotekniikan perusopinnot								
3Y2401	Opiskelutaidot ja ammatillinen kasvu	3op							
3T5000	Tietotekniikan perusteet	3op							
3T5090	Tietotekniikanoppimisen projekti	6op							
3Y2402	Tiedonhaku ja e-oppiminen		3op						
	Suomenkieli ja viestintä								
3Y2003	Kirjoittamisen ja esiintymistaidon kurssi				3op				
	Ruotsin kieli ja viestintä								
3Y2304	Ruotsin kieli ja viestintä								
	Englannin kieli ja viestintä								
3Y2301	English communication skills for engineering								
3Y2302	Business english for engineering								
	<b>Ammattiopinnot</b>								
	Sähkötekniikan perusopinnot								
3S0521	Sähköpiirien perusteet								

**Kuva 18. Verkkosivuston tulostus**

## 6. OHJELMISTON TESTAUS

Opinnäytetyön aiheena olleen ohjelmiston testauksessa ei varsinaisesti ollut käytössä mitään tiettyä testausohjelmaa, koska ohjelmiston testausta suoritettiin koko projektin ajan. Jokaisen osakokonaisuuden valmistuttua, kyseinen kokonaisuus testattiin huolella käytännössä.

Seuraavassa käyn hieman läpi muutamaa yleisesti käytössä olevaa ohjelmistontestaukseen käytettävää testausohjelmaa.

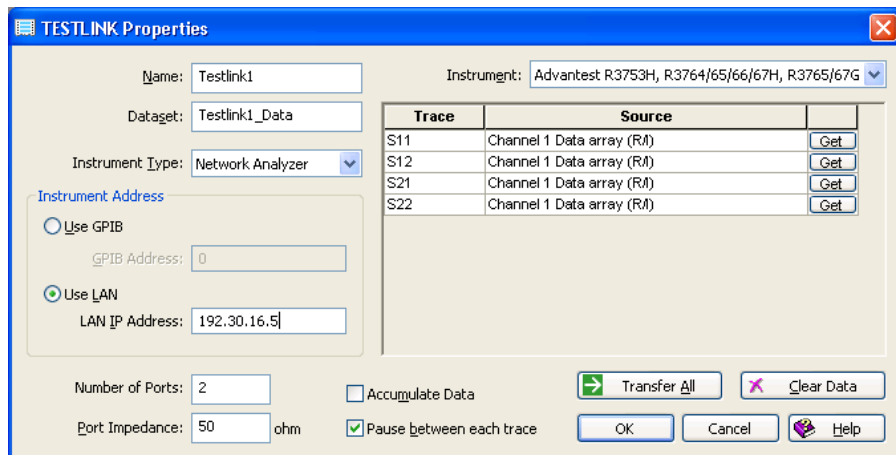
### 6.1. TestLink

TestLink on todella kattava, avoin testauksen hallintaan keskittyvä WWW-pohjainen työkalu. Järjestelmää on kehittämässä laaja kansainvälinen yhteisö, jolla ei ole kaupallisia riippuvuuksia. Ensimmäinen 1.0 versio julkaistiin 2005. Tällä hetkellä uusin vakaa versio on numeroltaan 1.7. Toiminnallisuuksiltaan TestLink keskittyy testauksen suunnittelun ja dokumentoinnin lisäksi testien raportointiin. Dokumentoinnin tavoitteena on noudattaa yleistä ohjelmistotestauksen standardia IEEE 829-1998, joka määrittää, millaista dokumentaatiota tarvitaan testauksen eri vaiheissa. Seuraavassa on listattu tärkeimmät Testlinkin ominaisuudet: /8/

- testitapausten dynaaminen kerääminen ja organisointi testisuunnitelmiksi sekä näiden kattava dokumentointi

- testitulosten sekä metriikkatietojen tarkastelu ja tulostaminen

- vaatimusten, testitapausten, julkaisujen sekä testitulosten välisen kahdensuuntaisen jäljitettävyyden ylläpito. Kuva 19. /8/



**Kuva 19. TestLink**

TestLink on rakennettu PHP:llä, mikä tekee siitä käyttöjärjestelmäriippumattoman. Se tukee MySQL, PostgreSQL ja Microsoft SQL Server-tietokantoja. Integraatorajapinta löytyy tunnettuihin virheen- ja tehtävähallintatyökaluihin Bugzilla, Mantis, Eventum, JIRA, Trac ja Track+. Testlinkin dokumentaatio ja raportit voidaan tulostaa HTML- ja Microsoft Office-muodoissa, jolloin ne ovat helposti toimitettavissa myös sellaisille henkilöille, joilla ei ole varsinaiseen järjestelmään pääsyä. Lisäksi järjestelmä tukee tiedon tuomisen (engl. *import*) muista järjestelmistä CSV- ja XML-muodoissa. Testlinkin käyttöliittymän ulkoasu on nähtävissä yllä olevassa kuvassa. /8/

## 6.2. BugZilla

Bugzilla on ilmainen Mozilla Foundationin kehittämä, avoimen lähdekoodin WWW-pohjainen virheidenhallintatyökalu. Järjestelmän kehitys alkoi 1998, ja sen jälkeen ehtinyt saavuttaa ohjelmistokehitysyhteisöjen ja -yritysten keskuudessa suuren suosion. BugZilla käyttäjäksi on ilmoittautunut muun muassa NASAn ja Nokian aliorganisaatioita. Järjestelmä on Mozilla Public License-lisenssin alainen, ja toteutettu CGI-pohjaisena Perlillä. Ympäristöltä Bugzilla-asennus vaatii vähintään WWW-palvelimen, Perl-tuen ja MySQL tai PostgreSQL-tietokantapalvelimen. Bugzilla keskittyy toiminnallisuudeltaan puhtaasti virheidenhallintaan. Perusasennus sisältää kaikki ominaisuudet mitä siihen tarvitaan eli virheiden raportoinnin, raporttien selaamisen ja haun sekä koosteraporttien luonnin. Perustoiminnallisuudet löytyvät jo käyttöliittymästä. Perustoimintojen ympärille on kehitetty

joukko ominaisuuksia, joilla järjestelmän käyttöä voidaan helpottaa ja osaksi myös automatisoida. Esimerkiksi virheiden korjausaikoja voidaan kirjata ja seurata, virheitä voidaan raportoida järjestelmään sähköpostitse, järjestelmä voidaan konfiguroida ilmoittamaan sähköpostitse raporttien muutoksista tai järjestelmä voidaan ajastaa lähettämään automaattisesti koosteraportteja sähköpostiin. Tietoja voidaan tarvittaessa siirtää myös muihin järjestelmiin, koska Bugzilla mahdollistaa myös virhelistojen tulostamisen järjestelmästä HTML-muodon lisäksi CSV-, XML- tai RDF - muodoissa. /4/

### **6.3. Ohjelmistokokonaisuuden testaus**

Tässä opinnäytetyössä testaus tapahtui, niin käyttöliittymän, tietokannan, kuin verkkosivustokokonaisuuden osalla, samanaikaisesti koodauksen edetessä. Eniten aikaa testauksessa meni, kun tietokannan yhteyttä sekä käyttöliittymän, että verkkosivuston tulostuksissa testattiin. Kaikki valikoiden sekä linkkien tulostukset toteutettiin suoraan tietokantahakuna ja tällä tavalla testaus oli hyvin järjestelmällistä sekä yksityiskohtaista.

Toisaalta aikaisemmin toteutettujen projektien perusteella todettakoon, että viimeinen ja ehkä paras testausmenetelmä edelleen lienee olevan loppukäyttäjät ja sovelluksen käyttö todellisissa tilanteissa.

Tästä johtuen, tämä opinnäytetyönä ollut sovellus, tuskin vielä tällaisenaan tulee toimimaan, vaan tähän tarvinnee tehdä parannuksia ja päivityksiä sitä mukaan, kun käyttökokeuksia oikeassa ympäristössä karttuu. Toisaalta edellä mainittujen testausohjelmien avulla tuskin kaikkia bugeja ohjelmistoista koskaan saadaan täysin karsittua pois. Paras testausympäristö siis edelleen lienee loppukäyttäjät.

## 7. POHDINTA/JATKOKEHITYSMAHDOLLISUUDET

Tämän opinnäytetyön tuloksena toteutettiin käyttöliittymä sekä tietokanta ja verkkosivustokokonaisuus Kemi-Tornion ammattikorkeakoulun opetussuunnitelmien päivittämistä ja ylläpito varten. Tähän kokonaisuuteen valittiin ohjelmointiympäristöksi QtSDK, jonka avulla C++-kielinen ohjelma on helposti toteutettavissa.

Tämän opinnäytetyön suunnittelu sekä toteutus vastasi oikeaa ohjelmistosuunnitteluprojektia. Aluksi mietittiin hyvin tarkkaan sovellukselle asetettuja vaatimuksia, joita sitten suunnittelun edetessä tarpeen mukaan tarkennettiin. Suunnitteluvaiheessa tutustuttiin tällä hetkellä käytössä olleeseen opetussuunnitelma verkkosivustoon ja siihen, mitä kaikkea parannuksia siihen olisi tarpeen tehdä. Alkuvaiheessa vastuukouluttajan kanssa pidetyissä palavereissa mietittiin lähinnä jo olemassa olevan sivuston puutteita sekä parannustarpeita. Kävimme alkuvaiheessa myös keskustelua siitä, millainen sovelluksen käyttöliittymä olisi mahdollisimman yksinkertainen mutta silti toimiva. Keskustelun tukena ja havainnollistamis-keinoina käytettiin yksinkertaisesti paperia ja kynää, joilla hahmoteltiin käyttöliittymän prototyyppejä.

Keskustelun pohjalta ja alan kirjallisuuteen tutustumalla punnittiin tarkoin sovelluksen ja sen käyttöliittymän toteutustapaa. Toteutustavaksi valikoitui ns. MainWindow-tyyppinen kokonaisuus, lähinnä sen Menu-valikoiden helppouden vuoksi. Qt-ympäristön valinta perustui siihen, että se on alustariippumaton, mikä mahdollistaa sovelluksen ajon melkein missä tahansa käyttöliittymässä, ilman sen kummempia muutoksia.

Koko ohjelmistokehityksen ajan pidettiin yhteyttä opinnäytetyön tilaajan suuntaan, tällä tavoin säännöllisesti tarkastellen sitä, kuinka ohjelmiston tarpeet tulee täytetyksi mahdollisimman kokonaisvaltaisesti.

Mielestäni tämän opinnäytetyön lopputuloksena valmistunut ohjelmisto vastaa kaikin puolin alussa asetettuja tavoitteita. Sovelluksella voidaan luoda kokonaan uusia opetussuunnitelmakokonaisuuksia, muokata jo olemassa olevia sekä päivittää ja ylläpitää kaikkia eri toimialojen ja koulutusohjelmien opetussuunnitelmia yhtä aikaa.

Nyt mietittynä tämä opinnäytetyöprojekti tarjosi minulle suunnattoman hyvät mahdollisuudet kehittyä ohjelmistosuunnittelussa. Varsinkin kun tein koko projektin kaikki vaiheet alusta loppuun yksin. Toisaalta ongelmien ilmaantuessa olisi toivonut, että taustalla olisi ollut tiimi, johon olisi voinut tarvittaessa tukeutua, mutta tässä kohtaa todettakoon se, että koulun opettajat olivat kiitettävästi tukemassa ongelmien ratkaisussa, silti antaen minun itse löytää ongelmiin ratkaisut.

Jatkokehitystä mietittäessä, uskon tästä sovelluksesta olevan mahdollisuus kehittää varsin pienellä vaivalla kokonaisvaltaisemmin palveleva konsepti ajatellen koulun tarvitsemia erilaisia tietokanta ja verkkosivusto kokonaisuuksia. Olen tässä hetkessä jo alustavasti miettinyt sitä, että tästä voisi alkaa kehittää sellaista, että tällä samalla sovelluksella voisi toteuttaa Kemi-Tornion ammattikorkeakoulun kompetenssianalyysit sekä muita koulun tarvitsemia analyyseja. Kokonaisuudessaan olen äärettömän tyytyväinen saavutettuun lopputulokseen ja parasta kaikessa, tämä ohjelmistokokonaisuus on mielestäni hyvä tarvittaessa laajentaa kattamaan suurempiakin tarpeita tulevaisuudessa.



## 8. YHTEENVETO

Yhteenvedona tästä opinnäytetyöstä voisi todeta sen, että kaikkiaan tämä projekti on toiminut juuri sillä tavoin kokonaisvaltaisen kehittävänä, kuin opinnäytetyöllä varmasti tarkoitus on toimia.

Omalla kohdallani kokemattomuus Qt-ympäristöstä, oli yksi suurimmista haasteiden tuojista, mutta tässä hetkessä on hyvä todeta se, että vaikka alkuvaiheessa kyseisen sovelluksen tarvitsemien SQL-ajurien asennusongelmien kanssa allekirjoittanut hetkittäin tuskastui ja oli jo muutamaan otteeseen luovuttamassa, niin silti varsinkin nuo kyseiset ongelmat ja niiden ratkaisu, toivat minulle todella paljon uutta tietoa, niin Linux-käyttöjärjestelmästä kuin Qt-ympäristöstäkin ja sen johdosta tämä kyseinen projekti onnistui mielestäni erinomaisesti.

Tämän kyseisen sovelluksen kohdalla toivoisin tässä hetkessä sitä, että tämä tulisi oikeaan käyttöön Kemi-Tornion ammattikorkeakoulussa, koska se osoittaisi viimeistään minulle sen, että tekemäni työ ei ole mennyt hukkaan. Toisaalta olen kuitenkin äärettömän tyytyväinen kokonaisuutena tähän opinnäytetyöhöni, sen aikaansaannoksiin ja toiminnallisuuteen.

Lopuksi kun vielä todetaan se, että tällä sovelluksella mielestäni on mahdollisuuksia laajennettavuuteen, tarpeen ilmentyessä, en kai voi muuta kuin olla tyytyväinen tekemisiini tämän projektin osalta. Tarkoitukseni olisi kehittää tätä sovellusta seuraavaksi siten, että tällä samalla kokonaisuudella voisi halutessaan laatia sekä yllä pitää laajempaa tietoa tietokannassa, samalla käyttäen tuota tietoa hyväksi myös verkkosivustolla. Eli näkisin, että tästä sovelluksesta olisi kehitettävissä kokonaisvaltaisesti palveleva sovellus esimerkiksi Kemi-Tornion ammattikorkeakoulun tarpeisiin.

Tulevaisuus näyttää, mitä tämän sovelluksen kohdalla käy, mutta itse ainakin suhtaudun luottavaisin mielin tulevaisuuteeni. Tämä opinnäytetyö osoitti minulle viimeistään sen, että olen kehittynyt ohjelmistonsuunnittelussa siinä määrin viime vuosina, että voin hyvillä mielin valmistuttuani siirtyä työelämän tarjoamiin haasteisiin.

## 9. LÄHDELUETTELO

- /1/ Haikala, Ilkka, UML-kaaviot, 2009, saatavilla WWW-muodossa <URL: <http://www.cs.tut.fi/~otupk/kesakalvot/kayttotapaukset.pdf/>>, viitattu 19.4.2012
- /2/ Kruhten P, The 4+1 View Model of Architecture, IEEE Software JNL, 2004.
- /3/ Kuutti, Wille, Linux, 1. painos, WSOY 2007.
- /4/ Mozilla Organization, *Bugzilla*, 2008, saatavilla WWW-muodossa <URL: <http://www.bugzilla.org/>>, viitattu 18.4.2012.
- /5/ Qt-peruskurssi, 2010, saatavilla WWW-muodossa <URL: [http://www.slideshare.net/juhapeltomaki/qt-peruskurssi /](http://www.slideshare.net/juhapeltomaki/qt-peruskurssi/)>, viitattu 10.4.2012.
- /6/ Qt-sovelluskehys, 2011, saatavilla WWW-muodossa <URL: <http://reaktor.fi/osaaminen/qt-sovelluskehys/>>, viitattu 12.3.2012.
- /7/ Taipaleenmäki, Sami, PHP&MySQL, 1. painos, WSOY 2004.
- /8/ TestLink Community, *TestLink – open source test management*, 2008, saatavilla WWW-muodossa <URL: <http://www.teamst.org/>>, viitattu 20.4.2012.
- /9/ TestLink, *TestLink – avoid bugs by testing*, 2008, saatavilla WWW-muodossa <URL: <http://testlink.org/wordpress/>>, viitattu 20.4.2012.