

TAMPEREEN AMMATTIKORKEAKOULU  
Tietotekniikan koulutusohjelma  
Ohjelmistotekniikka

Tutkintotyö

Jyri Vuorinen

**Tenttisovellus**

Työn ohjaaja  
Työn teettäjä  
Tampere 2006

DI Tony Torp  
Tampereen ammattikorkeakoulu, valvojana DI Tero Markkanen

# TAMPEREEN AMMATTIKORKEAKOULU

Tietotekniikka

Ohjelmistotekniikka

Vuorinen, Jyri

Tutkintotyö

Työn ohjaaja

Työn teettäjä

Helmikuu 2005

Hakusanat

TenttISOVELLUS

38 sivua

DI Tony Torp

Tampereen ammattikorkeakoulu, valvojana DI Tero Markkanen

Java, Swing, Servlet, Applet, MySQL, Sun Application Server

## TIIVISTELMÄ

TenttISOVELLUS tehtiin helpottamaan opetustyötä Tampereen ammattikorkeakoulussa. Sovellus on sekä opettajien että opiskelijoiden käytettävissä TAMK:n työasemien WWW-selaimilla. Opettajat voivat luoda ja hallita tenttejä, joita opiskelijat voivat suorittaa.

Työ toteutettiin Java-ohjelmointikielellä. Sovellus sijaitsee TAMK:n UNIX-palvelimen Sun Application Server -palvelinohjelmistossa. Sovelluksen tietovarastona käytetään MySQL-tietokantaa ja sovelluksen rajapinta Winha-järjestelmään (LDAP-hakemisto) on toteutettu Java Servlet -tekniikalla.

Työssä täytyi ottaa tietoturvariskit erityiseen tarkkailuun, koska sovelluksen käyttöliittymä suoritetaan asiakaskoneella, jolloin asiakas saa käsiinsä käyttöliittymän binääriluokat.

Sovellukseen kehitetään edelleen uusia ominaisuuksia ja tietoturvaa parannetaan hajauttamalla opettajan ja opiskelijan käyttöliittymät eri sovelluksiksi.

## TAMPERE POLYTECHNIC

Computer Engineering

Software Engineering

Vuorinen, Jyri

Engineering Thesis

Thesis Supervisor

Commissioning Company

February 2005

Keywords

Exam Software

38 pages

DI Tony Torp

Tampere Polytechnic, DI Tero Markkanen

Java, Swing, Servlet, Applet, MySQL, Sun Application Server

## ABSTRACT

Exam Software was made to help teaching work at Tampere Polytechnic. The software is available for teachers and students on every web browser enabled workstation at TAMK. Teachers can create and manage exams that students perform.

Work was made with Java language. The software is on TAMK UNIX server's Sun Application Server. The software has an interface to third-party databanks (Winha's LDAP-directory). Exam software's own data is stored in MySQL database. The Interface has been developed using Java Servlet technology.

Security issues had to be taken seriously, because software's graphical user interface is executed on client's workstation. Therefore client has access to GUI's binary classes.

Software is still being developed and security will improve by splitting teacher's and student's interfaces into two different applications.

## SISÄLLYSLUETTELO

<b>KÄYTETYT TERMIT JA LYHENTEET .....</b>	<b>6</b>
<b>1 JOHDANTO .....</b>	<b>7</b>
<b>2 TENTTISOVELLUS.....</b>	<b>7</b>
<b>3 KEHITYSYMPÄRISTÖ JA TYÖKALUT.....</b>	<b>8</b>
3.1 Java-ohjelmointikieli.....	8
3.2 Sun Application Server 8.1 -palvelinohjelmisto.....	8
3.3 NetBeans-sovelluskehitin .....	8
3.4 MySQL-tietokanta .....	9
3.5 Subversion-versionhallinta .....	9
<b>4 SOVELLUKSEN ARKKITEHTUURI .....</b>	<b>9</b>
4.1 Sovelluksen kuvaus.....	9
4.2 Pääelementit.....	10
4.3 Winha-rajapinta.....	12
4.4 Käyttöliittymän UML-luokkakaaviot .....	14
4.5 Projektin rakenne sovelluskehittämissä .....	15
4.6 Tietoliikenneyhteydet .....	17
4.7 Tietokanta .....	17
4.7.1 Tietokannan kuvaus .....	18
4.7.2 Transaktiot .....	18
4.7.3 Tietokannan käsitekaavio.....	20
<b>5 KÄYTTÖLIITTYMÄ .....</b>	<b>21</b>
5.1 Kuvaus .....	21
5.2 Käyttöliittymän apuluokat .....	22
5.2.1 Tietokanta .....	22
5.2.2 Tamk .....	23
5.2.3 Lajitteleva-taulukko .....	23
5.3 Opettajan käyttöliittymä.....	24
5.3.1 Kirjautuminen .....	25
5.3.2 Päävalikko.....	26
5.3.3 Omat tentit .....	26
5.3.4 Tenttieditori.....	27
5.3.5 Tentin opiskelijat .....	29
5.3.6 Tentin esikatselu .....	29
5.3.7 Tentin tarkastus.....	30
5.3.8 Julkiset tentit .....	31
5.3.9 Kurssin kysymykset .....	31
5.3.10 Kysymyseditori .....	32

5.3.11	Kysymyksen esikatselu.....	32
5.4	Opiskelijan käyttöliittymä.....	33
5.4.1	Kirjautuminen .....	33
5.4.2	Opiskelijan tentit.....	33
5.4.3	Tentin suorittaminen .....	34
5.4.4	Tentin tulokset .....	34
<b>6</b>	<b>SOVELLUKSEN TIETOTURVA JA JATKOKEHITYSSUUNNITELMAT.....</b>	<b>35</b>
6.1	Käyttäjätunnukset .....	35
6.2	Käyttöliittymän luokkien obfuskointi .....	35
6.3	Winha-rajapinnan tietoturva .....	36
6.4	Käyttöliittymän jakaminen kahteen osaan .....	36
6.5	Arvosanalaskuri .....	36
<b>7</b>	<b>LOPPUPÄÄTELMÄT JA YHTEENVETO .....</b>	<b>37</b>
	<b>LÄHDELUETTELO.....</b>	<b>38</b>

## KÄYTETYT TERMIT JA LYHENTEET

Asiakas	WWW-selaimella sovellusta käyttävä henkilö
FK	Foreign Key, vierasavain tietokannan taulussa
J2EE	Java 2 Platform, Enterprise Edition. Javan lisäosat mm. hajautettuihin järjestelmiin
Java	Java-ohjelmointikieli
JRE	Java Runtime Environment. JRE:n avulla voidaan ajaa Java-ohjelmia esim. WWW-selaimella
JVM	Java Virtual Machine, Java-virtuaalikone
Käyttäjä	Sovelluksen käyttäjäprofiili. Käyttäjäprofiileita ovat opettaja ja opiskelija.
LDAP	Lightweight Directory Access Protocol on hakemistopalvelujen käyttöön tarkoitettu verkkoprotokolla /1/
MySQL	Avoimeen lähdekoodiin perustuva tietokantaohjelmisto
Palvelin	Palvelinkone, jossa tenttisovellus sijaitsee fyysisesti
Palvelinohjelmisto	Tässä sovelluksessa käytetään palvelinohjelmistona Sun Application Server 8.1 Enterprise Edition -ohjelmaa
PK	Public Key, pääavain tietokannan taulussa.
SQL	Structured Query Language, SQL-kyselykieli
Winha	TAMK:n käyttämä opiskelijarekisteri
Winha-koodi	Yhdistelmä kurssin koodista ja toteutusnumerosta. Esim. S1234-1.

## 1 JOHDANTO

Tampereen ammattikorkeakoulussa on opiskelijoita noin 5000 ja vakinaista henkilökuntaa noin 400. /2/

Jokainen opiskelija suorittaa opiskeluaikansa useita kymmeniä tenttejä, joista suurin osa tehdään kynällä ja paperilla. Näin ollen opettajilta kuluu useita tunteja laatiessaan tenttejä ja tarkastaessaan opiskelijoiden suorittamia tenttejä. On syntynyt tarve sovellukselle helpottamaan opettajien työtä ja opiskelijoiden opintoja.

Sovelluksen suunnittelu alkoi keväällä 2005, jolloin suunnittelin tietokannan ja käyttöliittymän. Kesällä 2005 toteutin sovelluksen ensimmäisen version, jonka jälkeen jatkoin seuraavan version suunnittelua ja toteutusta. Samaisena kesänä myös tilattiin tietokonekeskukselta rajapinta sovellukselle. Kevään 2006 aikana sovelluksen versio 1.2 valmistui.

## 2 TENTTISOVELLUS

Tenttisovelluksen perusideana on päästä eroon paperitenteistä. Se helpottaa opettajia tenttien laatimisessa ja opiskelijoita tenttien suorittamisessa. Sovellus tarkastaa osan opiskelijoiden tenttivastauksista automaattisesti, jolloin opettajalle jäävät vain esseevastaukset tarkastettavaksi. Vastausten tarkastukseen on sovellukseen kehitetty oma työkalu. Opiskelijat saavat näin ollen tenttien tulokset entistä nopeammin ja kivuttomammin. Sovellus on helppokäyttöinen ja sitä voi käyttää miltä tahansa TAMK:n työasemalta ilman lisäohjelmien asennusta.

### 3 KEHITYSYMPÄRISTÖ JA TYÖKALUT

Tässä kappaleessa käydään läpi sovelluksessa käytetyt ohjelmointikielet ja työkalut.

#### 3.1 Java-ohjelmointikieli

Sovelluksen toteutuskieleksi valittiin Java-ohjelmointikieli, koska se tarjoaa erittäin hyvät käyttöliittymäkirjastot, joilla on helppo rakentaa käyttöliittymiä. Javalla tehdyt sovellukset ovat helposti siirrettävissä eri käyttöjärjestelmästä toiseen.

Sun Microsystems Inc:n kehittämä Java-ohjelmointikieli on yksinkertainen, oliopohjainen, arkkitehtuurista riippumaton, tehokas, verkkokäyttöön soveltuva ohjelmointikieli. /3/

#### 3.2 Sun Application Server 8.1 -palvelinohjelmisto

TAMK:n tietokonekeskus ehdotti palvelinohjelmistoksi Sun Application Server -ohjelmistoa. Palvelinohjelmistoksi olisi myös hyvin käynyt esimerkiksi Apache Tomcat.

Sun Microsystems Inc:n kehittämä palvelinohjelmisto tarjoaa Javan J2EE- ja J2EE-alustat palvelinpuolen ja Web-palveluiden sovelluksille. /4/

#### 3.3 NetBeans-sovelluskehitin

Projektin alussa käytin Borland JBuilder -sovelluskehittäjä, mutta kesällä 2005 vaihdoin sovelluskehittäjäksi NetBeans-ohjelman, koska se tarjoaa erinomaiset rajapinnat Sun Application Server -palvelinohjelmistolle sekä on riittävän monipuolinen projektin tarpeisiin.

NetBeans on Sun Microsystems Inc:n kehittämä sovelluskehitin. Se on graafisella käyttöliittymällä toteutettu kehitysympäristö J2SE- ja J2EE-sovellusten implementointiin ja se on suhteellisen monipuolinen ja yksinkertainen käyttää.

NetBeans, Sun Application Server, J2SE ja J2EE -komponentit on pakattu samaan pakettiin ja ne voi ladata suoraan Sun Microsystems Inc:n sivuilta.



### 3.4 MySQL-tietokanta

MySQL-tietokanta valittiin tähän projektiin, koska se on ilmainen ja sen ominaisuudet riittävät täysin hyvin projektin tarpeisiin.

MySQL on maailman suosituin avoimen lähdekoodin tietokantapalvelin, jota kehittää ruotsalainen MySQL AB -yritys. /5/

### 3.5 Subversion-versionhallinta

Projektissa käytettiin Subversion-versionhallintaa lähdekoodien versiointiin. Tämän avulla pystyin palaamaan versioissa taaksepäin tarvittaessa.

Subversion on avoimen lähdekoodin versionhallintaohjelmisto, jonka kehityksestä vastaa <http://subversion.tigris.org>.

## 4 SOVELLUKSEN ARKKITEHTUURI

Tässä luvussa kuvataan sovelluksen rakenne, sen käyttämät tietoliikenneyhteydet, tietokannan kuvaus ja sovelluksen siirrettävyys.

### 4.1 Sovelluksen kuvaus

Tenttisovellusta käytetään WWW-selaimella. Sovellus koostuu kahdesta pääelementistä, jotka ovat keskenään yhteydessä ja ottavat tarvittaessa tietoliikenneyhteydet järjestelmän ulkoisiin tietovarastoihin.

Tenttisovelluksen käyttöliittymä on toteutettu Javan Swing-käyttöliittymäkirjastolla. Käyttöliittymä on upotettu Javan Applet-sovelmaan, jolloin käyttöliittymän suoritus annetaan asiakkaan koneella toimivalle WWW-selaimelle. Selaimen tulee tukea Javan virtuaalimoottoria. Swing-kirjaston avulla saadaan käyttöön parempi käyttöliittymä verrattuna perinteiseen palvelimessa suoritettavaan käyttöliittymään.

Sitä vastoin tietoturva heikkenee asiakkaalla suoritettavissa sovelluksissa. Sovelluksen tietoturvaa käsitellen myöhemmin luvussa 5. *Sovelluksen tietoturva ja jatkokehityssuunnitelmat.*

Sovelluksen rajapintoja ei ajeta asiakkaan työasemalla, vaan palvelimen puolella.

Sovellus on siirrettävissä mihin tahansa alustalle, joka tukee Sun Application Server 8.1 -palvelinohjelmistoa sekä MySQL 4.1.12 -tietokantaa tai sitä uudempaa.

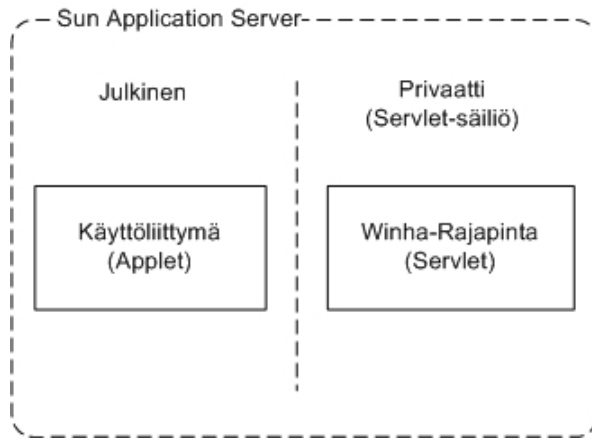
## 4.2 Pääelementit

Sovellus koostuu kahdesta pääelementistä, jotka ovat käyttöliittymän Java-luokat upotettuna Applet-sovelmaan ja Servlet-tekniikalla toteutettu rajapinta opiskelijarekisteriin eli ns. Winha-rajapinta.

- Winha-rajapinta  
Toimii sovelluksen rajapintana TAMK:n LDAP-kantaan. Suoritetaan palvelimella.
- Käyttöliittymä  
Asiakkaan näkymä sovellukseen. Suoritetaan asiakkaan tietokoneella.  
Käyttöliittymällä on yhteys tietokantaan, johon tallennetaan kaikki käsiteltävä data.

Sovelluksen Java-luokat ovat käytettävissä TAMK:n tietokonekeskuksen UNIX-palvelimella Sun Application Server Enterprise Edition 8.1-palvelinohjelmistossa. Samalla palvelimella sijaitsee myös käyttöliittymän käyttämä MySQL-tietokanta.

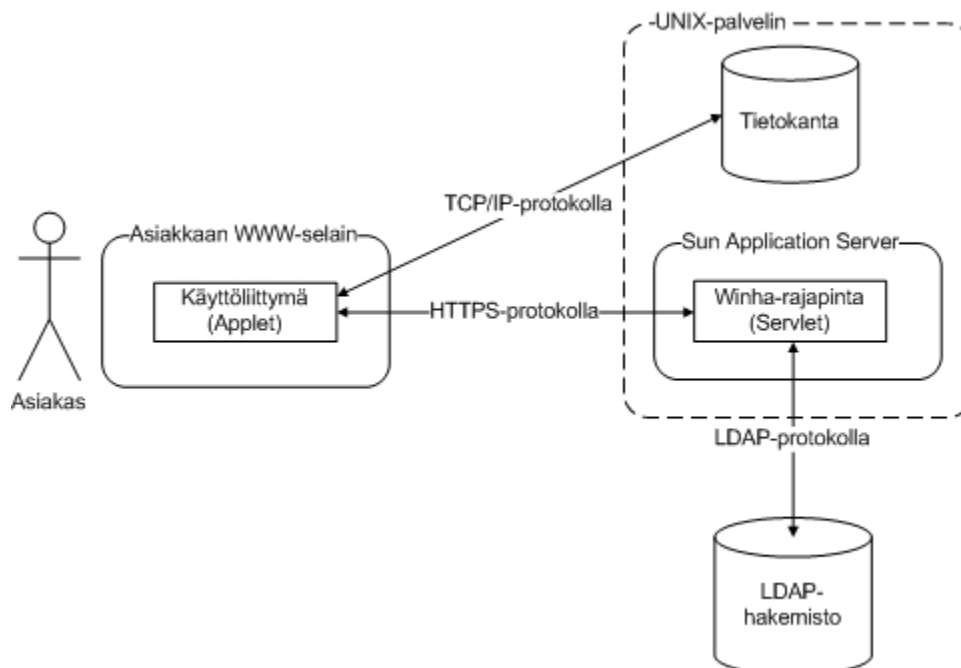
Seuraava kuva näyttää pääelementtien näkyvyyden palvelinohjelmiston sisällä.



**Kuva 1** Pääelementtien näkyvyys palvelinohjelmistossa

Käyttöliittymäelementti on julkisessa käytössä ja se saadaan ladattua WWW-selaimella asiakkaan työasemalle. Winha-rajapinta on taas privaattialueella ja siihen saa yhteyden vain käyttöliittymästä käsin salasanan avulla. Käyttöliittymää ei suoraan käytetä palvelimelta, vaan se siirtyy asiakkaan työasemalle yhteyttä pyydettyäessä.

Alla olevassa kuvassa esitetään koko sovelluksen arkkitehtuuri. Kuvassa asiakas on ladannut selaimeensa Applet-sovelmaan upotetut käyttöliittymäluokat.



**Kuva 2** Sovelluksen arkkitehtuuri

Asiakkaan WWW-selaimesta on tietoliikenneyhteydet tietokantaan ja Winha-rajapintaan. Winha-rajapinta toteuttaa tietoliikenneyhteydet ulkoiseen LDAP-hakemistoon.

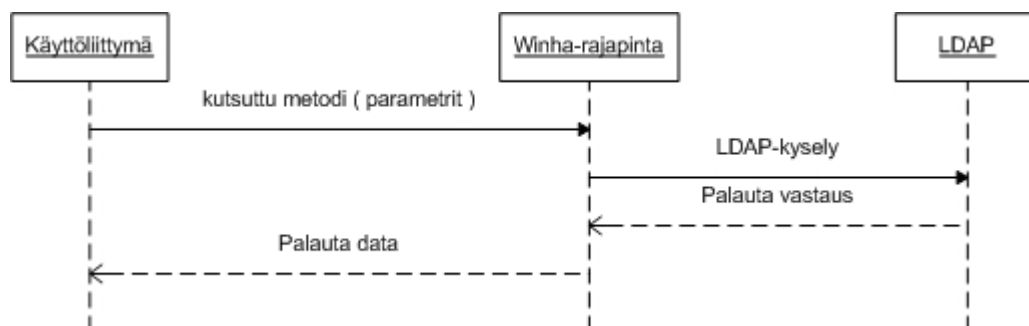
### 4.3 Winha-rajapinta

Rajapinta toimii Winhan rekisterin ja käyttöliittymän välissä. Se tarjoaa kaiken tarvittavan informaation opettajista, kursseista ja niiden opiskelijoista.

Tietoturvan vuoksi rajapinta päätettiin toteuttaa Java Servlet -sovelluksena, joka sijaitsee Sun Application Server -palvelinohjelmiston Servlet-säiliössä. Servlet-luokkiin ei pääse käsiksi säiliön ulkopuolelta.

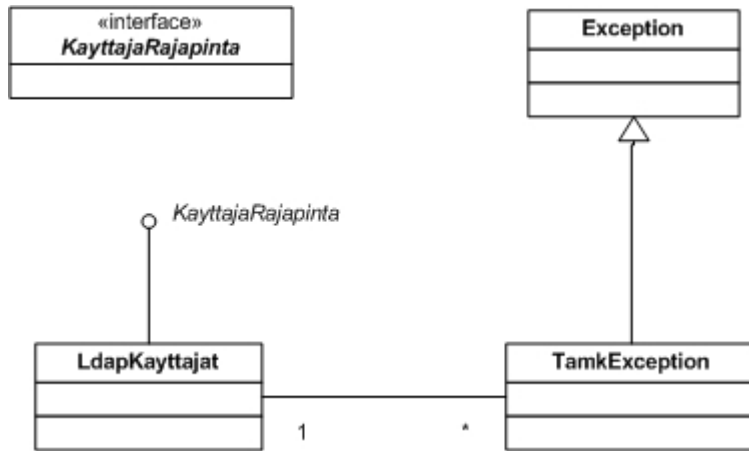
Käyttöliittymän Tamk-apuluokka ottaa rajapintaan HTTPS-yhteyden ja lähettää rajapintaan salasanan. Rajapinnan todettua salasanan oikeaksi avataan molemminpuolinen objektivirta. Käyttöliittymä kutsuu haluttua rajapinnan metodia objektivirran kautta ja rajapinta muodostaa LDAP-yhteyden opiskelijarekisteriin. Tämän jälkeen rajapinta palauttaa rekisteristä saapuneen datan käyttöliittymälle. Suunnittelin rajapinnan ja tilasin sen toteutuksen TAMK:n tietokonekeskukselta.

Seuraavassa sekvenssikaaviossa kuvataan rajapinnan toimintaa.



Kuva 3 Käyttöliittymä kutsuu rajapinnan metodeita

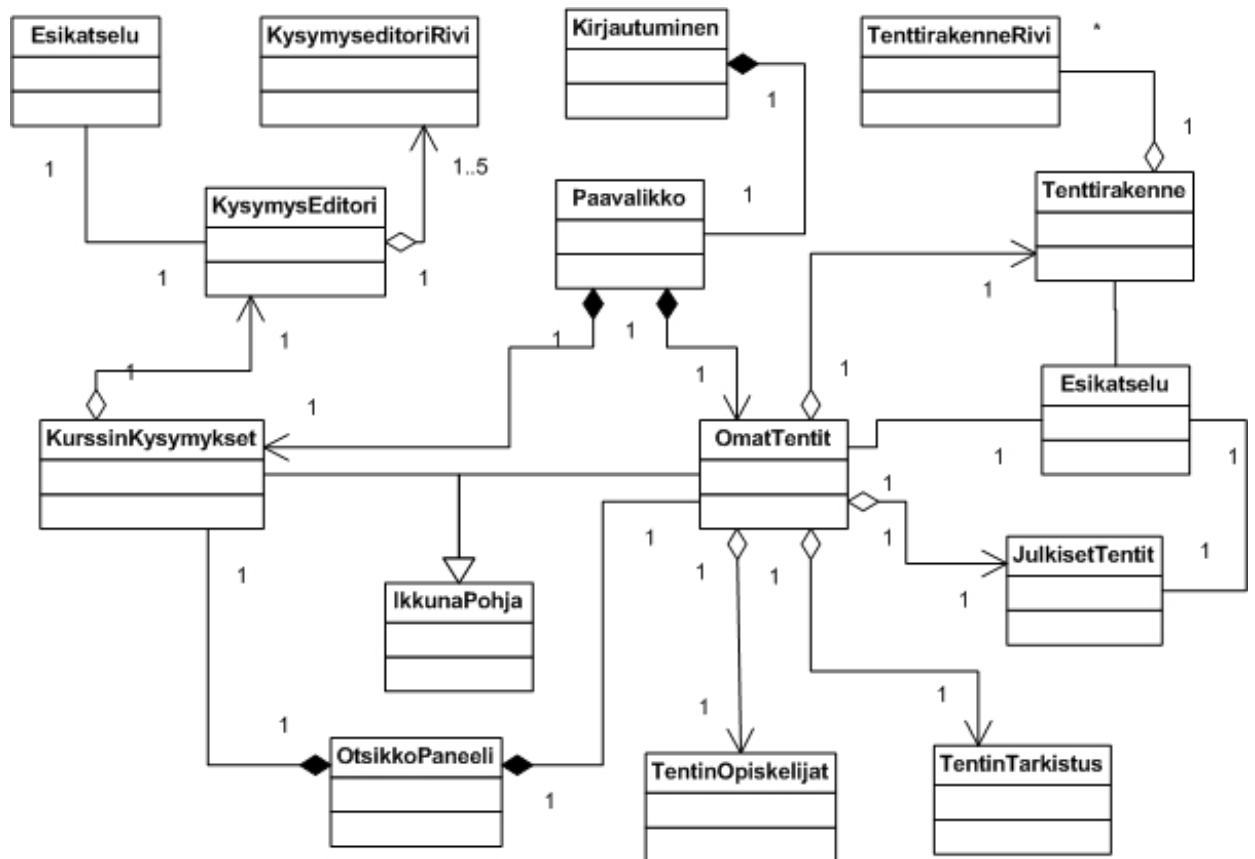
Alla oleva luokkakaavio näyttää rajapinnan rakenteen.



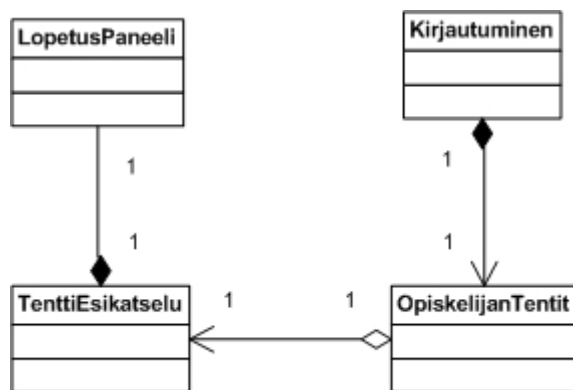
Kuva 4 Rajapinnan UML-luokkakaavio

#### 4.4 Käyttöliittymän UML-luokkakaaviot

Seuraavissa kuvissa näytetään opettajan ja opiskelijan käyttöliittymien UML-luokkakaaviot.



Kuva 5 Opettajan käyttöliittymän UML-luokkakaavio



Kuva 6 Opiskelijan käyttöliittymän UML-luokkakaavio

#### 4.5 Projektin rakenne sovelluskehittämissä

Sovelluskehittimellä ohjelmointi on järjestelmällisempää kuin esimerkiksi pelkällä tekstieditorilla. Sovelluskehitin tarjoaa useita työkaluja projektin automatisoinniksi. Tässä työssä tärkeimmät kehittimen ominaisuudet ovat sisäänrakennettu versionhallintarajapinta sekä Sun Application Server -palvelinohjelmistolle lähetettävien XML-asetustiedostojen ja WAR-paketin lähetys, jotka sovelluskehitin konfiguroi ja pakkaa automaattisesti.

Sovelluskehittämissä sovellus on jaettu kahteen projektiin:

- Tamk-hakemisto  
Pääprojekti, joka sisältää sovelluksen Servlet-luokkien lähdekoodit.
- Tenttikanta  
Sisältää käyttöliittymäkomponenttien ja niiden apuluokkien lähdekoodit. Tämä projekti liitetään pääprojektin juureen.

Projektien lähdekoodit käännetään NetBeans-sovelluskehittimellä, jolla ne myös lähetetään (deploy) palvelinohjelmistolle WAR-pakettina (zip-formaatti). Palvelinsovellus tämän jälkeen purkaa paketin palvelimelle.

Seuraavassa taulukosta näemme palvelimella sijaitsevan tenttisovelluksen hakemistorakenteen.

**Taulukko 1 Palvelinsovelluksessa sijaitsevan tenttisovelluksen tiedostot ja hakemistot**

Hakemisto / Tiedosto	Kuvaus
WEB-INF	Sisältää sovelluksen Servlet-luokat (mm. Winha-rajapinta). Pääsy vain WWW-palvelimen kautta, joten luokkien binääritiedostoja ei voi ladata asiakkaan koneelle, vaan ne suoritetaan palvelimella.
META-INF	Meta-tietoja luokista.
com	MySQL-tietokannan tiedostoja mm. ajurit.
csv_files	Oppilaiden tulokset tallennetaan tänne väliaikaisesti CSV-tiedostoina.
esikatselu	Käyttöliittymän pakkaus. Sisältää esikatselun luokat.
kirjautuminen	Käyttöliittymän pakkaus. Sisältää kirjautumiseen käytetyt luokat.
kysymys	Käyttöliittymän pakkaus. Sisältää kysymysnäkyymiin liittyviä luokkia.
lajiteltavataulukko	Käyttöliittymän pakkaus. Sisältää näkymissä käytetyn taulukon luokkia.
opiskelija	Käyttöliittymän pakkaus. Opiskelijanäkymän luokat löytyvät täältä.
org	MySQL-tietokannan tiedostoja.
paavalikko	Käyttöliittymän pakkaus. Päävalikon luokat ovat täällä.
tentti	Käyttöliittymän pakkaus. Tenttien hallintaan liittyvät luokat sijaitsevat tässä pakkauksessa.
tietokanta	Käyttöliittymän pakkaus. Tietokantayhteydet muodostava luokka on täällä.
yleiset	Käyttöliittymän pakkaus. Apuluokkia käyttöliittymän muille pakkauksille.
kirjautuminen.jsp	Avaa käyttöliittymän kirjautumissovelluksen.
index.jsp	Avaa kirjautuminen.jsp-linkin uuteen ikkunaan.



## 4.6 Tietoliikenneyhteydet

Tenttisovelluksen käyttöliittymään otetaan HTTP- tai HTTPS-yhteys WWW-selaimella, joka tukee Javan virtuaalimoottoria (JVM). JVM mahdollistaa sovelluksen ajon selaimessa. Yhteyttä pyydetessä lataa selain sovelluksen käyttöliittymän luokat – jotka näytetään Applet-sovelmassa – käyttäjän selaimen muistiin.

Taulukosta 2 nähdään sovelluksen käyttöön vaadittavat avoimet tietoliikenneportit.

**Taulukko 2 Palvelimelle vaadittavat avoimet TCP/IP-yhteydet käyttöliittymää käytettäessä**

Yhteys	Portti
HTTP-yhteys palvelimelle	8080
HTTPS-yhteys palvelimelle	8181
MySQL-tietokantayhteys	3306

Taulukossa 3 näytetään palvelinsovelluksen käyttöön vaadittava avoin tietoliikenneportti.

**Taulukko 3 Palvelimelle vaadittava avoin TCP/IP-yhteys hallintaliittymää käytettäessä**

Yhteys	Portti
Sun Application Server -palvelinohjelmiston	4849
WWW-hallintasovellus	

Sovelluksessa käytettävällä signeeraamattomalla Applet-sovelmalla ei ole oikeutta ottaa yhteyttä muuhun palvelimeen kuin siihen, josta se ladattiin käyttäjälle. Sovelluksen luokkien täytyy siis sijaita samalla palvelimella kuin tietokanta. Applettien signeerausta ei ole mahdollista käyttää TAMK:n työasemissa, koska se vaatisi käytännössä jokaiseen työasemaan muutoksen Applettien tietoturvapoliikkaan (Java policy). /6/

## 4.7 Tietokanta

Tietokannaksi valittiin MySQL-tietokanta, koska se on ilmainen, vakaa ja sen ominaisuudet riittävät täysin hyvin projektin tietovarastoksi.

#### 4.7.1 Tietokannan kuvaus

Sovelluksen informaation säilömiseen käytetään MySQL-tietokannan versiota 4.1.12-standard. Tietokannan kaikki taulut ovat InnoDB-tyyppiä, joka on hitaampi kuin perinteinen MyISAM-tyyppi, mutta InnoDB-tyyppi tukee taulujen viite-eheyttä.

InnoDB-tyyppi tuli osaksi MySQL:n versiossa 3.23.34a, ja siitä tuli oletustyyppi lähtien versiosta 4.0. /7/

Tietokantaan tallennetaan opettajien laatimat kysymykset, tentit, kurssitiedot ja opiskelijoiden suoritteet.

#### 4.7.2 Transaktiot

Suurin osa sovelluksen SQL-kyselyistä on turvallista ajaa tietokantaan automaattisesti (auto commit), joka on myös oletuksena java.sql.Connection-luokassa. Aika ajoin kuitenkin sovelluksessa käytetään useita SQL-transaktioita yhtä tapahtumaa kohti. On erittäin tärkeää ohjelman toiminnan kannalta, että kaikki nämä kyselyt onnistuvat. Tämän takia osassa kyselyitä käytetään manuaalisia transaktioita, jolloin varmistetaan, että kaikki halutut kyselyt onnistuvat. Jos yksikin kysely epäonnistuu, niin mitään muutoksia tauluihin ei tehdä.

Alla on koodiesimerkki manuaalisista transaktioista käyttäen sovelluksen Tietokanta-apuluokkaa.

```
Tietokanta db = new Tietokanta(); // Luodaan uusi yhteys tietokantaan
Statement stmt = db.createStatement();
java.sql.Connection con = db.getConnection(); // Haetaan yhteysolio
try
{
    con.setAutoCommit(false); //Otetaan manuaaliset transaktiot käyttöön

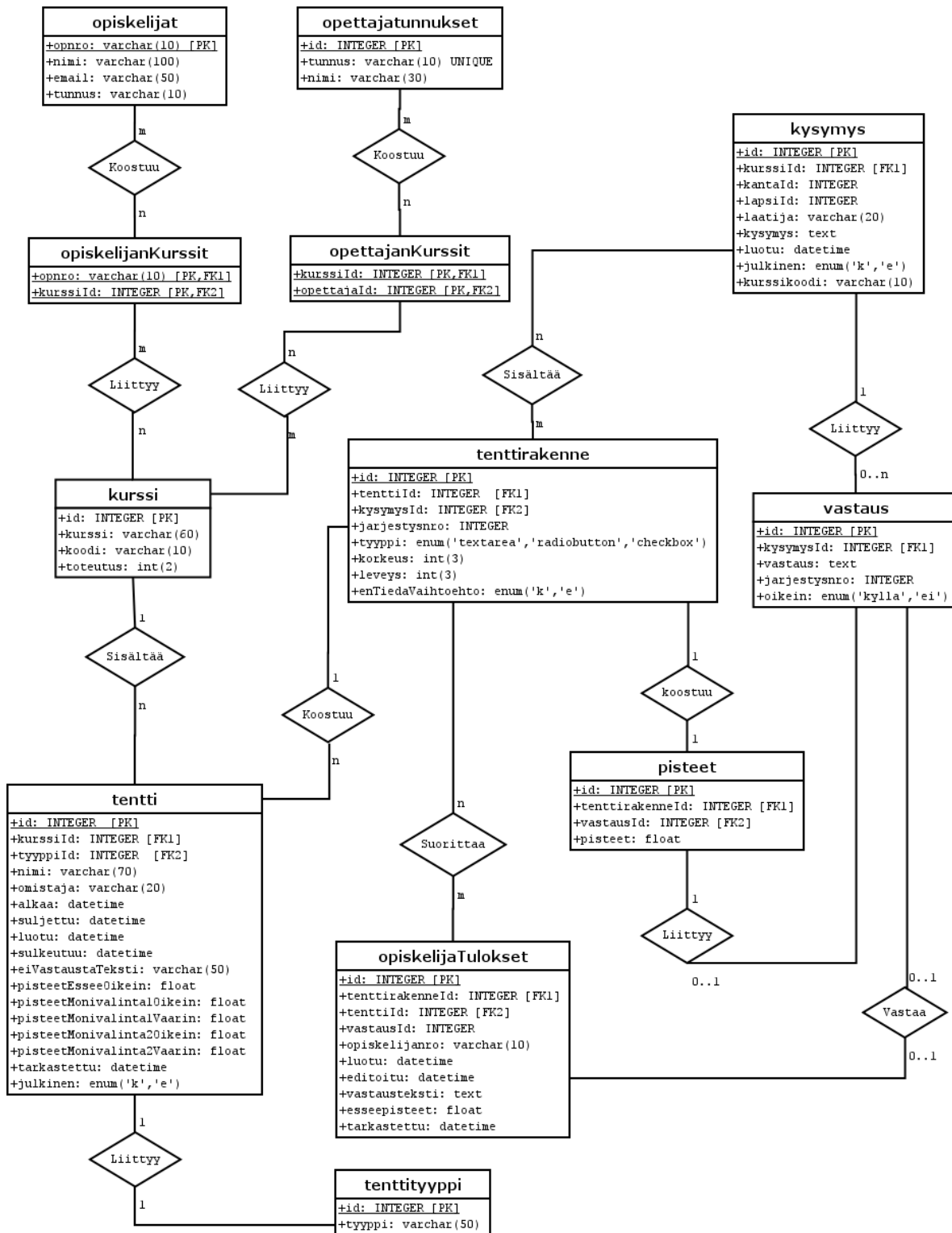
    /* Suoritetaan SQL-kyselyitä tietokantaan */

    stmt.executeUpdate("DELETE FROM esimerkkitaulu WHERE id="+bar);
    stmt.close();
    stmt.executeUpdate("INSERT INTO foo values ('bar')");

    /*
    * Jos kaikki SQL-kyselyt onnistuivat, niin toteutetaan halutut
    * muutokset tietokantaan commit()-metodilla.
    */
    con.commit();
}
catch(SQLException e)
{
    /*
    * Tähän tullaan jos kyselyistä yksikin epäonnistui. Suoritetaan rollback()-metodi,
    * jolloin haluttuja SQL-kyselyitä ei suoriteta kantaan.
    */
    con.rollback();
    e.printStackTrace(); //Tulostetaan syntynyt virhe.
}
db.disconnect(); //Suljetaan tietokantayhteys.
```

**Koodiesimerkki 2 Esimerkki tietokannan transaktioista**

4.7.3 Tietokannan käsitekaavio



Kuva 7 Tenttikanta-tietokannan käsitekaavio

## 5 KÄYTTÖLIITTYMÄ

Tässä kappaleessa käydään läpi käyttöliittymän rakenne ja sen näkymien toiminnot.

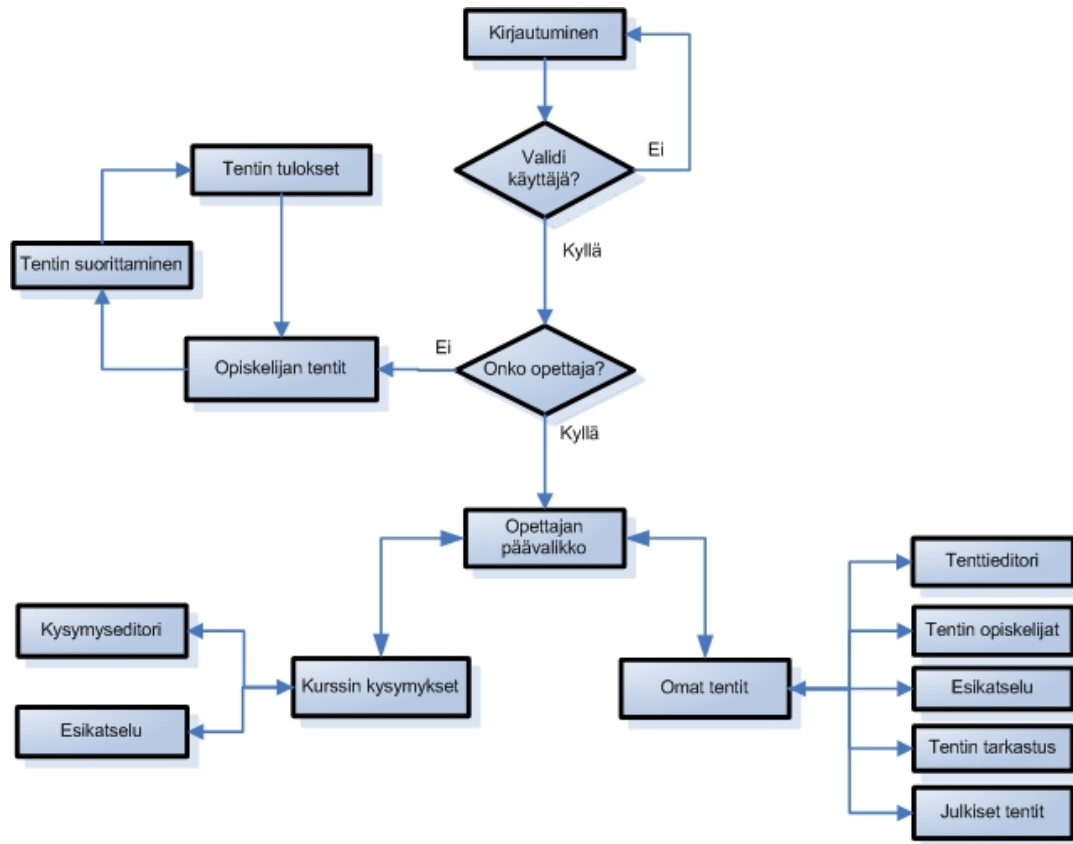
### 5.1 Kuvaus

Käyttöliittymä on sovelluksen käyttäjälle näkyvä osa. Sen suunnitteluun kulutettiin paljon aikaa. Uusia komponentteja lisättiin käyttöliittymään ja vanhoja muokattiin hieman. Käytettävyyttä mietittiin paljon, koska ihmisillä on eri tottumuksia ja tapoja käyttää käyttöliittymää. Oli aika haastavaa yrittää tehdä käyttöliittymästä kaikille mahdollisimman mieleinen käyttää. Sen käytettävyys pyrittiin tekemään mahdollisimman helppokäyttöiseksi ja suoraviivaiseksi.

Käyttöliittymä toteutettiin Javan Swing-käyttöliittymäkirjastolla sen tarjoamien monipuolisten käyttöliittymäkomponenttien vuoksi. Käyttöliittymää ei tehty perinteisellä palvelimen puolella (server-side) ajettavana ns. dynaamisen HTML:n generaattorina (JSP, Servlet, PHP tms.), koska sillä ei pysty toteuttamaan niin monipuolisia käyttöliittymiä, jolloin käytettävyys kärsii.

Käyttöliittymän lähtöpisteenä on kirjautumisnäkyvä sovellukseen. Kirjaututtaessa käyttöliittymä ohjautuu opettajan näkymään tai opiskelijan näkymään riippuen käyttäjän käyttäjäryhmästä.

Alla oleva kulkukaavio kartoittaa käyttöliittymän päänäkymät.



Kuva 8 Käyttöliittymän kulkukaavio

## 5.2 Käyttöliittymän apuluokat

Käyttöliittymässä on näkymättömiä osia, joita kutsun käyttöliittymän apuluokiksi. Näitä luokkia käyttöliittymä kutsuu suorittamaan haluttuja toimintoja.

### 5.2.1 Tietokanta

Tämä luokka hoitaa käyttöliittymän tietokantayhteydet.

Kun käyttöliittymässä luodaan instanssi tästä luokasta, niin Tietokanta-luokan rakentajassa otetaan yhteys tietokantaan. Luokalla on muutamia metodeita, joita käyttämällä pysyy

käyttöliittymän lähdekoodi paljon siistimpänä, eikä koodia täydy uudelleen kirjoittaa jokaiseen tietokantakyselyyn.

### 5.2.2 Tamk

Luokka ottaa yhteydet Winha-rajapintaan ja kutsuu sen metodeita. Tamk-luokka käsittelee rajapinnalta saadun datan ja siirtää sen käyttöliittymälle.

### 5.2.3 Lajitteleva-taulukko

Tätä luokkaa käytetään näkymissä, jossa tarvitaan informaation taulukointia.

Näkymät koostavat linkitetty listat näytettävästä informaatiosta ja lähettävät taulukon otsikkorivien nimikkeiden kera lajittelevalle taulukolle. Tämän jälkeen taulukkoluokka palauttaa instanssiin näkymälle. Näkymä liittyy palautetun instanssin sille varatulle paneelille.

Alla on lyhyt esimerkkikoodi taulukon käytöstä.

```
LinkedList tentit = haeData(); //Haetaan tentit linkitettyyn listaan
if(tentit != null)
{
    // Luodaan taulukolle otsikkorivi
    String []taulukonOtsikot = { "Tentin nimi", "Tunnus", "Tila" };

    // Luodaan instanssi taulukosta
    // Parametrina annetaan tentit, taulukon otsikot ja tämän olion instanssi.
    LajittelevaTaulukko taulukko = new LajittelevaTaulukko( tentit, taulukonOtsikot, this );

    // Asennetaan taulukko paneeliin
    setTaulukko(taulukko);
}
```

**Koodiesimerkki 1 Lajittelevantaulukon käyttöesimerkki**

### 5.3 Opettajan käyttöliittymä

Opettajan käyttöliittymässä hallitaan pääasiassa tenttejä ja niihin liittyviä kysymyksiä.

Tentissä voi olla kolmentyyppisiä kysymyksiä:

- Yhden oikean vastauksen monivalintakysymys (radiobutton group)
  - Vastauksia voi määritellä 2-5, joista vain yksi voi olla oikea vaihtoehto.
  - Tähän kysymystyyppiin voidaan myös liittää ”Ei vastausta” -vaihtoehdon, jonka pistearvo nolla, joka on vakio.
- Usean oikean vastauksen monivalintakysymys (checkbox group)
  - Vastauksia voi määritellä 2-5, joissa voi olla useita oikeita vaihtoehtoja.
- Esseekysymys
  - Esseekysymyksellä ei ole vastausvaihtoehtoja, vaan tekstikenttä, johon vastaus kirjoitetaan.

Kaikille kysymyksille voidaan tentissä asettaa oikeiden vastauksen pistemäärät.

Monivalintakysymyksille voi lisäksi myös määritellä väärin vastauksien pistemäärät.

Kysymykset on suunniteltu erilleen varsinaisesta tentistä, joten kysymyksiä voi luoda ilman tenttiä, ja samaa kysymystä voidaan käyttää useassa tentissä. Kysymykset eivät suoraan myöskään viittaa sen vastausvaihtoehtoihin, vaan kysymyksen vastaus viittaa kysymykseen. Tämän avulla kysymyksen tyyppiä voidaan muuttaa tekemättä uutta kysymystä. Esimerkiksi jos kysymys on suunniteltu monivalintakysymykseksi ja sillä on viisi eri vastausvaihtoehtoa, voidaan kysymyksen tyyppi muuttaa tentissä esseekysymykseksi tekemättä kysymyksestä kopiota.

Uuden tentin luonti on kaksivaiheinen. Ensin asetetaan tentin toteutus, tyyppi, aukeamis- ja sulkeutumisaika. Toisessa vaiheessa poimitaan tenttiin valmiit kysymykset ja/tai luodaan



uusia kysymyksiä. Kysymykset tulevat tenttiin päällekkäin ja niiden järjestystä voidaan haluttaessa muuttaa. Käyttäjä pisteyttää oikeat ja väärät vastaukset, joko kysymys kerrallaan tai kerralla koko tentin kaikki kysymykset kysymystyyppin mukaan. Kysymykset voidaan pisteyttää kokonais- tai liukulukuarvoilla.

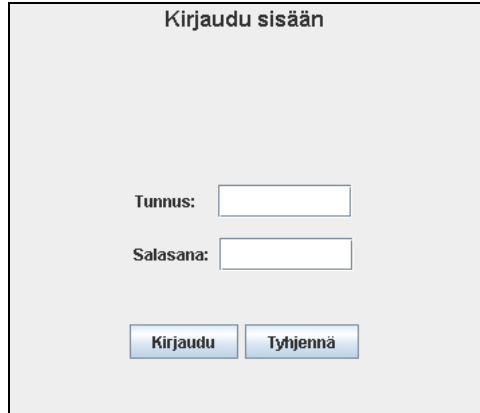
Oman toteutuksen tenttejä voi myös julkaista julkiseksi, jolloin muut käyttäjät voivat kopioida tentin itselleen.

### 5.3.1 Kirjautuminen

Käyttäjän ladattua käyttöliittymän luokat selaimen on kirjautuminen ensimmäinen näkymä asiakkaalle.

Käyttäjän autenttisuus tarkastetaan Winha-rajapinnassa. Käytettävä tunnus on sama, jolla käyttäjät kirjautuvat esimerkiksi koulun Windows-työasemille. Kun rajapinta toteaa sille syötetyn tunnuksen ja salasanan oikeiksi, kysyy sovellus rajapinnalta, mihin käyttäjäryhmään syötetty tunnus kuuluu.

Opettaja on henkilö, joka on Tampereen ammattikorkeakoulun henkilökuntaa ja hän on opettajana vähintään yhdellä toteutuksella. Opettaja voi luoda tenttejä ja kysymyksiä kurseilleen, avata ja sulkea tenttejä, lisätä jälkikäteen opiskelijoita toteutukselle ja tarkastaa opiskelijoiden esseevastaukset (monivalintavastaukset sovellus tarkastaa automaattisesti). Jokainen tentti ja sen kysymykset kuuluvat aina jollekin kurssin toteutukselle, joten tämän tenttejä voi hallita yksi tai useampi opettaja, joka on kyseisellä toteutuksella opettajana. Kysymykset taas näkyvät kaikille kurssin toteutuksille, joten niitä voi esimerkiksi kopioida toteutukselta toiselle. Opettaja voi kirjautua vain tenttien hallintaan.



Kirjaudu sisään

Tunnus:

Salasana:

Kuva 9 Käyttäjän kirjautumisnäky

### 5.3.2 Päävalikko

Päävalikosta voi siirtyä omiin tentteihin, kurssien kysymyksiin ja kirjautua ulos järjestelmästä.



Käyttäjä: testi

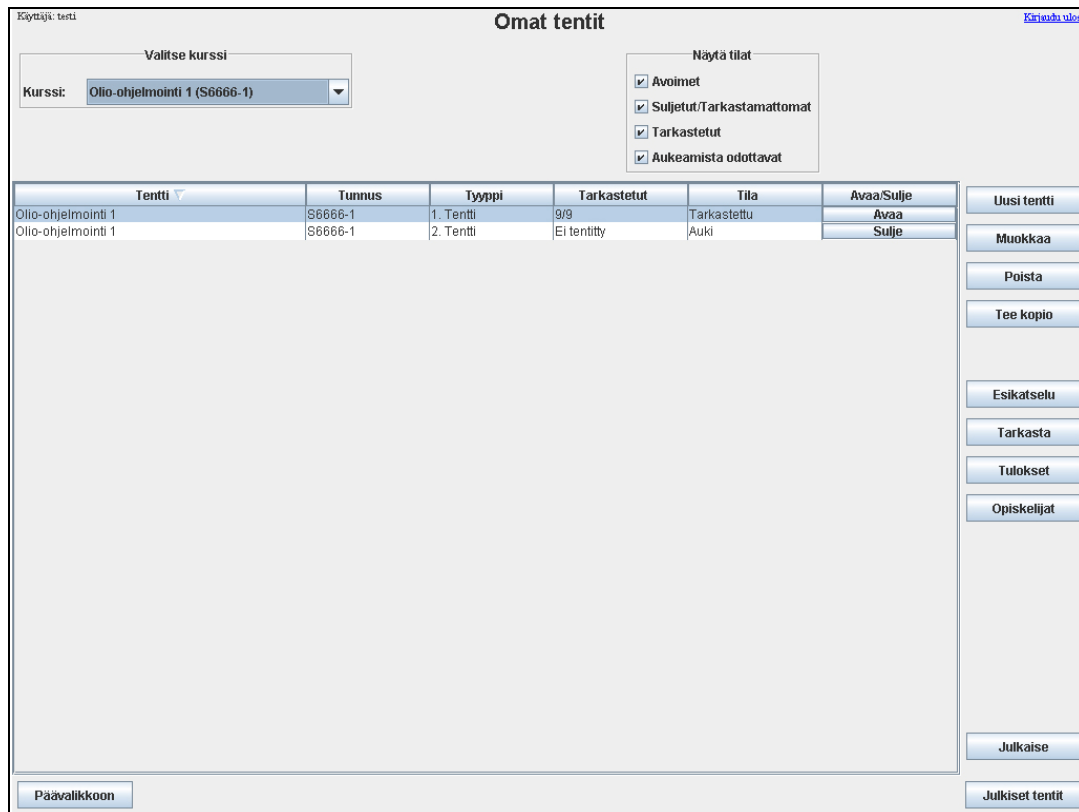
Päävalikko

[Kirjaudu ulos](#)

Kuva 10 Opettajan päävalikkonäkymä

### 5.3.3 Omat tentit

Tämä on tenttien hallintatyökalu. Olemassa olevat tentit näytetään taulukossa. Tenttejä voi esikatsella, muokata, poistaa, tarkastaa, tehdä kopioita, avata ja sulkea opiskelijoille, julkaista näkyväksi kaikille kurssin toteutuksille, ladata tarkastetun tentin tulokset Microsoft Excel -ohjelmaan CSV-tiedostona, lisätä opiskelijoita kursseille tai luoda uusia tenttejä.



Kuva 11 Opettajan näkymä tenttien hallintaan

### 5.3.4 Tenttieditori

Uutta tenttiä luotaessa annetaan tentille ensin kurssin tyyppi, tentin tyyppi, alkamis- ja sulkeutumisaika.



Kuva 12 Uuden tentin asetukset

Kuvassa 12 määritellään tentin asetukset. Tämä on ensimmäinen vaihe tenttiä luotaessa.

Seuraava vaihe on rakentaa itse tentin rakenne valmiista kysymyksistä. Tämän jälkeen tentin kysymykset pisteytetään ja tentti voidaan tallentaa.

Käyttäjä: testi

Olio-ohjelmointi 1 (S6666-1). 1. Tentti. 1.1.2006 12:00 - 1.1.2006 13:00

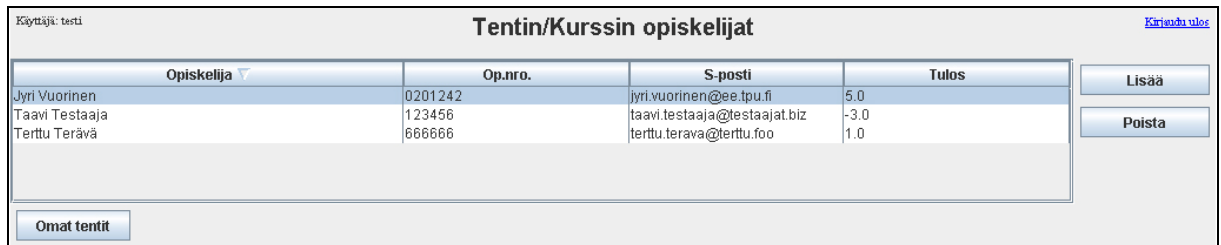
Siirrä	Numero	Kysymys	Asetukset	Poista
<input type="checkbox"/>	1	Onko C++ -kielessä oliota?	<input type="button" value="Asetukset"/>	<input type="button" value="Poista"/>
<input type="checkbox"/>	2	Mikä seuraavista väittämistä on oikein?	<input type="button" value="Asetukset"/>	<input type="button" value="Poista"/>
<input type="checkbox"/>	3	Mitä tarkoittaa C++ -kielessä varattu sana public?	<input type="button" value="Asetukset"/>	<input type="button" value="Poista"/>

Kuva 13 Tenttiin on liitetty kolme kysymystä

Tenttiä tallennettaessa tietokantaan tallentuvat tentin tiedot ja rakenne. Rakenteeseen kuuluvat tenttiin valitut kysymykset, vastaukset ja pisteet. Tietokantaan tallentuvat tässä vaiheessa myös toteutuksen opiskelijoiden tiedot, jotka haetaan Winha-rajapinnasta.

### 5.3.5 Tentin opiskelijat

Tässä näkymässä listataan tentin opiskelijat. Opiskelijoita voi lisätä ja poistaa tenttiin täältä.



Käyttäjä: testi [Kirjaudu ulos](#)

Opiskelija	Op.nro.	S-posti	Tulos
Jyri Vuorinen	0201242	jyri.vuorinen@ee.tpu.fi	5.0
Taavi Testaaja	123456	taavi.testaaja@testaajat.biz	-3.0
Terttu Terävä	666666	terttu.terava@terttu.foo	1.0

Kuva 14 Tentin opiskelijat

### 5.3.6 Tentin esikatselu

Tentin esikatselun voi avata tenttiä luodessa sekä omat tentit -näköymästä. Opiskelijan suorittama tentti näyttää täsmälleen samalta kuin esikatselussa näkyvä tentti.



Olio-ohjelmointi 1. 1. Tentti. 1.1.2006

1. Onko C++ -kielessä oliota? (Oikea vastaus 1.0 p.; Väärä vastaus -1.0; 'Ei vastausta' -vastaus 0 p.)

Kyllä  
 Ei  
 Ei vastausta

2. Mikä seuraavista väittämistä on oikein? (Oikea vastaus 1.0 p.; Väärä vastaus -1.0 p.)

C-kielessä on try-catch -lohko.  
 Bjarne Stroustrup kehitti C++-kielen.  
 Java-kielessä on varattusana boolean.  
 1+1=5

3. Mitä tarkoittaa C++ -kielessä varattu sana public? (Maksimipisteet 5 p.)

Kuva 15 Tentin esikatselu

### 5.3.7 Tentin tarkastus

Opettaja tarkastaa ja pisteyttää suoritettujen tenttien esseevastaukset. Seuraavassa kuvassa näkyy kahden oppilaan tarkastamattomat esseevastaukset.

Opiskelija	Op.Nro	Tila
Taavi Testaaja	123456	Tarkastamaton
Terttu Terävä	666666	Tarkastamaton

Kuva 16 Esseevastausten tarkastus

Seuraava vaihe on tarkastaa esseekysymykset. Kuvassa 16 näkyy kahden opiskelijan esseevastaus. Opettaja voi pisteyttää vastaukset tässä.

( 1 / 1 ) Kysymys: Mitä tarkoittaa C++-kielessä varattu sana public?

4 / 0 Jotain julkista vissiin?

2 / 0 En muista

Kuva 17 Yhden esseekysymyksen vastausten pisteytys

Tämän jälkeen vastauspisteet tallennetaan ja voidaan siirtyä seuraavan kysymyksen vastausten tarkasteluun.

### 5.3.8 Julkiset tentit

Julkaistut tentit näkyvät kaikille sen kurssin toteutuksille. Täältä voi kopioida toisten toteutusten julkisia tenttejä omiin tentteihin.

Tentti	Koodi	Julkaisija
Virtapiirit	S1122	1. Tentti

Kuva 18 Julkisten tenttien näkymä

### 5.3.9 Kurssin kysymykset

Kurssin kysymykset näkyvät kaikille sen toteutuksille. Kysymyksiä ei siis julkaista samalla tavalla kuin tenttejä.

Kysymys	Tyyppi	Luotu
Onko C++-kielessä olioita?	radiobutton	24.04.2006 11:38
Mikä seuraavista väittämistä on oikein?	checkbox	24.04.2006 11:41
Mitä tarkoittaa C++-kielessä varattu sana public?	textarea	24.04.2006 11:43

Kuva 19 Kurssien kysymysten hallintatyökalu

### 5.3.10 Kysymyseditori

Kysymyseditorissa voi luoda essee- tai monivalintakysymyksiä. Vastauksia voi olla 1-5 kappaletta.

The screenshot shows a web form titled "Kysymyksen lisäys" (Add Question). At the top left, it says "Kirjautunut: testi" and at the top right, there is a "Kirjautu ulos" link. The form contains the following elements:

- Aihe:** A dropdown menu.
- Kurssi:** A dropdown menu with the value "Olio-ohjelmointi 1 (S6666)".
- Kysymyksen tyyppi:** A dropdown menu with the value "Monivalintakysymys (useampi oikea vastaus)".
- Kysymys:** A text input field containing "Mikä seuraavista väittämistä on oikein?".
- Vastaus 1:** A dropdown menu with "Väärin" selected, followed by a text input field containing "C-kielessä on try-catch-lohko".
- Vastaus 2:** A dropdown menu with "Oikein" selected, followed by a text input field containing "Bjarne Stroustrup kehitti C++-kielen".
- Vastaus 3:** A dropdown menu with "Oikein" selected, followed by a text input field containing "Java-kielessä on varattu sana boolean".
- Vastaus 4:** A dropdown menu with "Väärin" selected, followed by an empty text input field.
- Vastaus 5:** A dropdown menu with "Väärin" selected, followed by an empty text input field.

On the right side of the form, there are two buttons: "Tallenna" (Save) and "Esikatselu" (Preview).

Kuva 20 Uuden kysymyksen luonti

### 5.3.11 Kysymyksen esikatselu

Kysymyksiä voi esikatsella tällä luokalla.

The screenshot shows a preview window titled "Kysymyksen esikatselu" (Preview Question). At the top right, there is a "Sulje" (Close) button. The main content area displays the question text and four radio button options:

- Mikä seuraavista väittämistä on oikein?
- C-kielessä on try-catch-lohko
- Bjarne Stroustrup kehitti C++-kielen
- Java-kielessä on varattu sana boolean
- 1+1=5

Kuva 21 Kysymyksen esikatselu



## 5.4 Opiskelijan käyttöliittymä

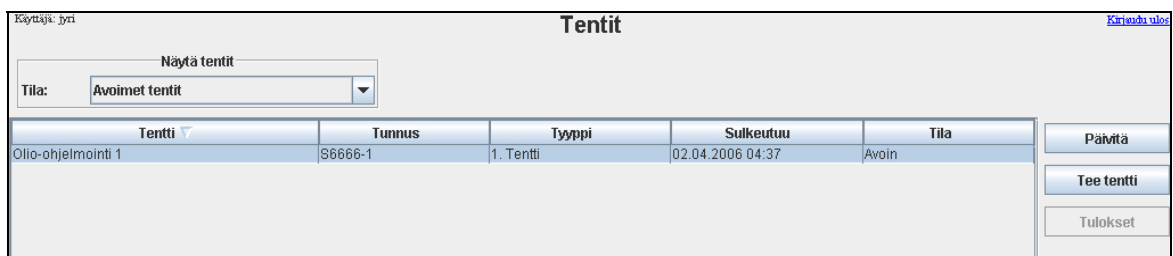
Opiskelija näkee vain omien kurssiensa avoimet tentit, jotka hän pystyy avaamaan suoritettaviksi.

### 5.4.1 Kirjautuminen

Opiskelija on henkilö, joka opiskelee Tampereen ammattikorkeakoulussa, ja hänellä on tunnukset TAMK:n työasemille. Opiskelija voi suorittaa vain hänen kurssien tenttejä, jotka toteutuksen opettaja on merkinnyt avoimiksi. Opiskelijan kirjautuessa onnistuneesti, siirtyy käyttöliittymän näkymä opiskelijan tentteihin. Opiskelija voi kirjautua vain ja ainoastaan opiskelijasovellukseen.

### 5.4.2 Opiskelijan tentit

Kun opiskelija on tehnyt tentin sovellus tarkastaa monivalintavastausten pisteet automaattisesti ja jättää esseevastausten tarkastamisen opettajalle. Opiskelija ei voi enää tehdä tenttiä uudelleen ellei opettaja avaa tätä hänelle, jolloin vanhat pisteet häviävät.



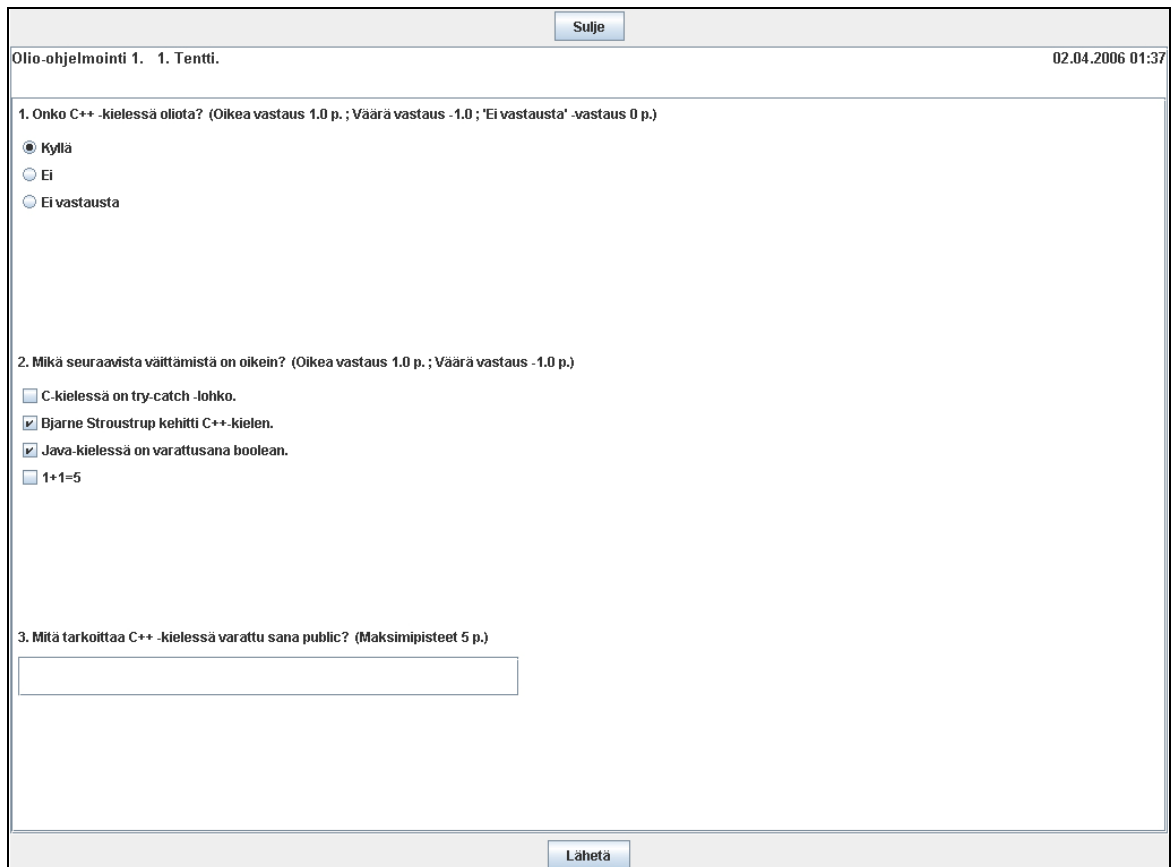
The screenshot shows a web interface titled 'Tentit'. At the top left, it says 'Käyttäjä: jyri'. In the center, there is a 'Näytä tentit' section with a dropdown menu labeled 'Tila:' set to 'Avoimet tentit'. Below this is a table with columns: 'Tentti', 'Tunnus', 'Tyyppi', 'Sulkeutuu', and 'Tila'. The table contains one row: 'Olio-ohjelmointi 1', 'S6666-1', '1. Tentti', '02.04.2006 04:37', and 'Avoin'. To the right of the table are three buttons: 'Päivitä', 'Tee tentti', and 'Tulokset'.

Tentti	Tunnus	Tyyppi	Sulkeutuu	Tila
Olio-ohjelmointi 1	S6666-1	1. Tentti	02.04.2006 04:37	Avoin

Kuva 22 Opiskelijan avoimet tentit

### 5.4.3 Tentin suorittaminen

Opiskelija vastaa tentin kysymyksiin ja lähettää tulokset tarkastettavaksi, jolloin käyttöliittymän näkymä siirtyy tentin tuloksiin.

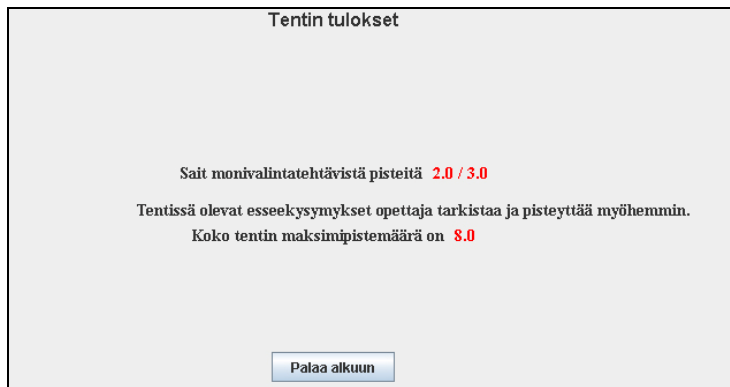


The screenshot shows a web browser window with a title bar containing a 'Sulje' button. The page content is titled 'Olio-ohjelmointi 1. 1. Tenti.' and includes a timestamp '02.04.2006 01:37'. The first question asks if the user is a C++ programmer, with radio buttons for 'Kyllä', 'Ei', and 'Ei vastausta'. The second question asks for correct statements, with checkboxes for 'C-kielessä on try-catch -lohko.', 'Bjarne Stroustrup kehitti C++-kielen.', 'Java-kielessä on varattuna boolean.', and '1+1=5'. The third question asks for the meaning of the 'public' keyword, with a text input field below it. A 'Lähetä' button is at the bottom.

**Kuva 23** Opiskelija on suorittamassa tenttiä

### 5.4.4 Tentin tulokset

Jos tentissä on monivalintakysymyksiä niin tarkastetaan nämä heti automaattisesti ja niiden tulosten summa näytetään opiskelijalle välittömästi. Mahdolliset esseevastaukset jätetään opettajan tarkastettavaksi ja tästä ilmoitetaan myös opiskelijalle.



Kuva 24 Tentin päätyttyä tulokset näytetään opiskelijalle

## 6 SOVELLUKSEN TIETOTURVA JA JATKOKEHITYSSUUNNITELMAT

Sovelluksen suunnittelussa täytyi kiinnittää erityistä huomiota tietoturvaan verrattuna palvelimella suoritettavaan sovellukseen. Suurimpana tietoturvariskinä on käyttöliittymä, koska se ajetaan asiakkaan työasemalla.

Seuraavissa aliotsikoissa käydään läpi keskeisimpiä tietoturvaratkaisuja.

### 6.1 Käyttäjätunnukset

Ilman käyttäjätunnusta ja salasanaa järjestelmään ei voi kirjautua.

Käyttäjien salasanaja ei tallenneta mihinkään. Ainoa käyttö salasanalle on kirjautumisen yhteydessä, jolloin rajapinnalta kysytään käyttäjän autenttisuus.

### 6.2 Käyttöliittymän luokkien obfuskointi

Asiakkaan selaimella suoritettavan Applet-sovelman ongelmana on se, että tenttisovelluksen käyttöliittymäluokat ovat asiakkaan työaseman kiintolevyllä, josta ne pystytään aukikoodaamaan (reverse engineer). Javan luokkia voi aukikoodata esimerkiksi siihen tarkoitukseen tehdyllä Jad-ohjelmalla (<http://www.kpdus.com/jad.html>). Avatun luokan

lähdekoodin näkee selkokielisenä, eli koodi on täysin luettavissa normaalina tekstinä, mukaan lukien String-tyyppisten muuttujien arvot, jolloin esimerkiksi tietokannan salasana on vaarassa joutua väärin käsiin.

Tätä ongelmaa helpottaa luokkien obfuskointi, jonka tarkoituksena on vaikeuttaa avatun luokan lähdekoodin lukemista. Tenttisovelluksen lähdekoodin obfuskointiin on käytetty JSshrink-ohjelmaa (<http://www.e-t.com/jshrink.html>). Obfuskoitua lähdekoodia on erittäin vaikea lukea ja ymmärtää. On tärkeää huomioida, että lähdekoodin obfuskointi ei estä lähdekoodin lukemista.

### **6.3 Winha-rajapinnan tietoturva**

Rajapintaan otetaan salattu HTTP-yhteys, eli HTTP-over-SSL-yhteys. Rajapinta vaatii salasanan sitä kutsuvalta sovellukselta.

Luokkien aukikoodauksen takia rajapinta ei ole osana käyttöliittymää vaan se on sijoitettu turvaan palvelinsovelluksen Servlet-säiliöön, jolloin ei ole mahdollista ladata rajapinnan luokkia palvelimelta. Rajapintaan otetaan yhteyttä Tenttisovelluksen Tamk-apuluokalla.

### **6.4 Käyttöliittymän jakaminen kahteen osaan**

Tietoturvan parantamiseksi sovellus jaetaan kahteen osaan, jolloin opiskelijaliittymän luokat sijaitsevat eri paikassa kuin opettajaliittymän. Opiskelijaliittymälle tehdään myös oma tietokanta, jolloin aukikoodatusta opiskelijaliittymästä ei pääse käsiksi kuin sen hetkisiin avoimiin tentteihin.

### **6.5 Arvosanalaskuri**

Tenttiin voi asettaa arvosanapisteraajat, jolloin suoritetuista pisteistä automaattisesti saadaan arvosana opiskelijalle.

## 7 LOPPUPÄÄTELMÄT JA YHTEENVETO

Tenttisovellus oli mielenkiintoinen ja haastava toteuttaa, koska tämänlaista sovellusta ei tietääkseni ole tehty aiemmin eikä täten ollut mitään mallia, josta ottaa esimerkkiä.

Suunnittelua riitti paljon, koska sovellukseen keksittiin uusia lisäominaisuuksia aika-ajoin ja vanhoja ominaisuuksia muutettiin.

Rajapinnan valmistuminen viivästyi useita kuukausia, joten tämän käyttöliittymään integrointitoimenpiteet pysähtyivät hetkeksi. Sen takia tein väliaikaisen staattista dataa palauttavan tynkäluokan rajapinnasta (ei siis ota yhteyttä LDAP-hakemistoon). En saanut tietokonekeskukselta MySQL-kantaan root-tunnusta. Tämä aiheutti myös hieman ongelmia sovelluksen kehityksessä, koska tietokanta aika-ajoin jostain syystä ajettiin alas enkä näin ollen pystynyt testaamaan tenttisovellusta koulun järjestelmissä. Asensin tämän takia kotiin omaan Linux-palvelimeen vastaavan tenttisovellusympäristön (Sun Application Server -ohjelmistopalvelin, MySQL-tietokanta ja Subversion-versionhallinta), jossa käytin tekemääni rajapinnan tynkäluokkaa. Tämä nopeutti kehitystä huomattavasti, koska pystyin tekemään kotona töitä eikä tullut ongelmia sovellusten oikeuksien kanssa.

## LÄHDELUETTELO

- [1] Yleistä tietoa LDAP-hakemistosta  
URL: <http://fi.wikipedia.org/wiki/LDAP>
  
- [2] Tampereen ammattikorkeakoulun sivusto  
URL: <http://www.tamk.fi/servlet/sivu/0/250559>
  
- [3] Yleistä tietoa Java-ohjelmointikielestä  
URL: <http://java.sun.com/docs/overviews/java/java-overview-1.html>
  
- [4] Yleistä tietoa Sun Application Server -palvelinohjelmistosta  
URL: <http://www.sun.com/software/products/appsrvr/index.xml>
  
- [5] Yleistä tietoa MySQL-tietokannasta  
URL: <http://www.mysql.com/why-mysql/>
  
- [6] Tietoa Java Applet -tekniikan signeerauksesta  
URL: <http://java.sun.com/developer/technicalArticles/Security/Signed/>
  
- [7] Tietoa MySQL-tietokannan InnoDB-taulutyypistä  
URL: <http://dev.mysql.com/doc/refman/4.1/en/innodb-in-mysql-3-23.html>