

Suunnitteludokumentti

Kivireki

Helsinki 13.11.2007

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Kurssi

581260 Ohjelmistotuotantoprojekti (6 ov)

Projektiryhmä

Anu Kontio
Ilmari Helen
Olli Juvonen
Joonas Murtola
Teppo Niinimäki

Asiakas

Timo Aalto

Ohjaaja

Jari Suominen

Kurssin vastuhenkilö

Kimmo Simola

Kotisivu

<http://www.cs.helsinki.fi/group/kivireki>

Versiohistoria

Versio	Päiväys	Tehdyt muutokset
0.1	?10.2007	Pohja luotu
0.2	3.11.2007	Arkkitehtuuri-osio lisätty
0.3	4.11.2007	Tarkastusvalmis versio
1.0	13.11.2007	Jäädetytty versio.

Sisältö

1 Johdanto	1
2 Sanasto	2
2.1 Käytettävien tekniikoiden nimet	2
2.2 Yleiset termit	3
3 Arkkitehtuuri	4
3.1 Osajärjestelmät	4
3.2 Komponentit	4
3.2.1 UI (User Interface)	4
3.2.2 Business	6
3.2.3 ORM (Object-relational mapping)	6
3.2.4 MySQL	6
3.2.5 Tomcat	7
4 Tietokanta	8
4.1 Käyttäjä	8
4.2 Kirja	9
4.3 Kurssi	9
4.4 Kurssikerta	9
4.5 Kurssikerran kirja	10
4.6 Laitos	10
4.7 Tila	11
4.8 Rooli	11
5 Tiedostot	12
5.1 Business-luokat	12
5.2 Model-luokat	12
5.3 Action-luokat	12
5.4 Kartoitustiedostot (konfiguraatitiedostot)	12
5.5 JSP-sivut	13
5.6 CSS-tiedostot	13

6 Luokkien kuvaukset	14
6.1 Luokkakaavio	14
6.2 JSP-sivut	14
6.2.1 login.jsp	15
6.2.2 coursemanagement.jsp	15
6.2.3 lecturer.jsp	16
6.2.4 library.jsp	17
6.3 CSS-tyylitiedostot	18
6.4 Java-luokat	18
A Java-luokat	19

1 Johdanto

Tämä on suunnitteludokumentti syksyn 2007 Ohjelmistotuotantoprojekti-kurssilla toteutettavalle kurssikirjatietojen hallintajärjestelmälle. Järjestelmän avulla opetushallinto ja luennoitsijat voivat koota tiedot järjestettävistä kursseista sekä niiden kurssikirjoista ja välittää tiedot kirjastolle. Kirjaston henkilökunta pystyy järjestelmän avulla hotamaan kurssikirjojen hankintaan ja yleiseen hallintaan liittyviä toimenpiteitä. Järjestelmä on alkuvaiheessa tarkoitettu Kumpulan tiedekirjaston ja Helsingin yliopiston Tietojenkäsittelytieteen laitoksen yhteiseen käyttöön. Myöhemmin sitä on mahdollista laajentaa koskemaan Helsingin yliopiston laitoksia ja kirjastoja.

Järjestelmä tullaan asentamaan kirjaston palvelimelle ja sitä käytetään WWW-selaimella. Ohjelmisto toteutetaan Java-kielellä ja tietojen tallennukseen käytetään MySQL-tietokantaa. Toteutus tehdään MVC-mallin mukaisesti ja siinä hyödynnetään Apache Struts - ja Hibernate -sovelluskehysä.

Tämä dokumentti kuvaa toteutettavan järjestelmän sellaisella tasolla, että toteutus on suoraviivainen. Johdannon ja sanaston lisäksi dokumentti on jaettu neljään osioon: Arkkitehtuuri, Tietokanta, Tiedostot, Luokkien kuvaukset. Käyttöliittymäsuunnitelma on vaatimuskäytännössä, joten sitä ei käydä kovinkaan tarkasti läpi. Arkkitehtuuri-osiossa selvitetään suhteellisen yleisesti järjestelmän korkean tason rakenne. Arkkitehtuuriosan Osajärjestelmät-aliotsikon alla kerrotaan, millaisista kokonaisuuksista järjestelmä koostuu ja kuvataan niiden roolia. Sitten käydään läpi järjestelmän komponentit ja niiden toiminta ja käyttö. Tämän jälkeen kuvataan yksityiskohtaisesti käytettävä tietokannan rakenne ja listataan taulujen luomiseen tarvittavat komennot. Lopuksi on kuvaus toteutuksen luokkarakenteesta sekä tarkempi luettelo kaikista luokista ja niiden tiedoista.

2 Sanasto

2.1 Käytettävien tekniikoiden nimet

Apache Struts – Web-sovelluksille tarkoitettu sovelluskehys, joka perustuu sovelluksen toimintalogiikan jakamiseen MVC-mallin mukaiseksi. Struts abstrahoi J2EE:n Servlet rajapinnan ja helpottaa sovelluksen toteutuksen rutiininomaisten puuhien toteutuksessa.

Apache Tomcat – Web-palvelinohjelmisto, joka toteuttaa Javan Servlet-rajapinnan.

Hibernate – Avoimen lähdekoodin sovelluskehys tietokantaoperaatioita varten.

HQL – Hibernate Query Language on Hibernaten oma tietokanta-kyselykieli, jonka avulla tietokantoihin voidaan tehdä mm. hakuja ja muutoksia. Katso myös *SQL*.

InnoDB – Tietokannan tallennusmoottori, joka tuo hyödyllistä lisätoiminnallisuutta tietokantaoperaatioihin.

JavaScript – Web-selaimen oliopohjaisen komentosarjakieli.

JDBC – Java Database Connectivity, Javan tietokantarajapinta, joka määrittelee tavan jolla Java-sovellus voi ottaa yhteyden tietokantaan. Sovelluksen ja tietokannan välisen kommunikoinnin hoitaa käytännössä tietokantakohtainen tietokanta-ajuri.

JSP – JavaServer Pages on Javan Servlet-rajapintaa hyödyntävä tekniikka, joka mahdollistaa web-sivujen dynaamisen generoinnin selainohjelmille. JSP käyttää omaa merkkkausmenetelmäänsä, joka mahdollistaa Java-koodin upottamisen HTML:n sekaan.

LDAP – Lightweight Directory Access Protocol, hakemistopalvelujen tietojen hallintaan tarkoitettu, TCP/IP:n päällä toimiva sovelluskerroksen verkkoprotokolla.

MVC – Model-View-Controller on ohjelmistotuotannossa käytettävä arkkitehtuurimalli, jossa ohjelmisto jaetaan kolmeen osaan: tietosisällön hallinta, käyttöliittymä sekä näitä hallinnoivaan kontrolliosio.

MySQL – Avoimen lähdekoodin olio-relaatiotietokantajärjestelmä. MySQL-kantaa voidaan hallinnoida SQL-kielellä.

SQL – Structured Query Language on tietokanta-kyselykieli, jonka avulla tietokantoihin voidaan tehdä mm. hakuja ja muutoksia.

XHTML – eXtensible Hypertext Markup Language on kuvauskieli, jota käytetään web-sivujen luomiseen. Erona perinteiseen HTML:n, XHTML täyttää XML-standardin muotovaatimukset.

2.2 Yleiset termit

arvioitu osallistujamäärä – Arvio kurssille osallistuvista opiskelijoista, johon uusien kirjojen hankinta suurelta osin perustuu. Tämä tieto syötetään järjestelmään laitoksen päässä. Kurssihallinto syöttää samanlaisen arvion myös TKT-laitoksen Ilmojärjestelmään.

kirjasto – Kirjasto on järjestelmän tilaaja sekä toinen pääasiallinen käyttäjä. Kirjasto käyttää järjestelmää kurssikirjatietojen hankintaan.

kurssi – Säännöllisesti tai vähemmän säännöllisesti järjestettävä opintokokonaisuus. Kurssi voidaan luennoida/järjestää useita kertoja. Jokaisella kurssilla on kurssikoodi (kurssin yksilöivä tunnus) sekä nimi. Esimerkiksi “581326-3, Java-ohjelmointi”, “99501Käyt, English Academic & Professional Skills: Reading, Writing & Spoken Communication” tai “477628, PSY382 Varianssianalyysi”.

kurssihallinto – Kurssihallinnolla tarkoitetaan dokumentissa opetushallintoa.

kurssikerta – Kurssin yksi järjestämiskerta. Esimerkiksi “Java-ohjelmointi, syksy 2007, periodi 2”.

laitos – Laitoksella viitataan järjestelmän toiseen käyttäjään – joko Tietojenkäsittelytieteen laitokseen tai yleisesti johonkin Helsingin yliopiston laitokseen. Laitoksen piiriin kuuluvat luennoitsija sekä kurssihallinto, jotka syöttävät järjestelmään tarpeelliset kurssi- ja kurssikirjatiedot.

sessio – Istunto. Yhtenäinen ajanjakso, jolloin käyttäjä on kirjautuneena järjestelmään.

tietokanta – “Kokoelma tietoja, joilla on yhteys toisiinsa.” Tässä dokumentissa tietokannalla viitataan käytettävään MySQL-relaatiotietokantaan.

3 Arkkitehtuuri

Järjestelmän arkkitehtuuri perustuu MVC-suunnittelumalliin. Sen ajatus on erottaa käyttöliittymä varsinaisesta sovelluslogiikasta ja datasta. Mallin edut perustuvat hyvään ylläpidettävyyteen. Mallin toteuttavassa järjestelmässä käyttöliittymää on helppo muuttaa ja järjestelmän osien vaihtaminen helpottuu.

Toinen järjestelmän arkkitehtuuriratkaisuista on kerrosarkkitehtuuri. Järjestelmä koostuu kolmesta kerroksesta. Ylimpänä on käyttöliittymäkerros, jonka alla on liiketoimintakerros. Alimpana on tietokantakerros. Kerrosarkkitehtuurimainen toteutus jakaa jokaiselle kerrokselle selkeät tehtävät. Tämä helpottaa järjestelmän ylläpitoa ja ymmärtämistä. Toisistaan riippumattomat kerrokset voidaan vaihtaa helposti toisiinsa. Esimerkiksi tietokanta voidaan vaihtaa tekemättä muutoksia käyttöliittymään.

3.1 Osajärjestelmät

Järjestelmä jakautuu MVC-mallin mukaisesti kolmeen osajärjestelmään:

- Model(Malli) hoitaa järjestelmän kaiken toimintalogiikan sekä tietojen tallentamisen tietokantaan ja niiden käsittelyn.
- View(Näkymä) määrittelee käyttöliittymän ulkoasun ja mallin tietojen esitystavan käyttöliittymässä.
- Controller(Ohjain) vastaanottaa käyttäjältä tulevat pyynnöt, ohjaa pyynnön mallille ja näyttää mallilta tulevan vastauksen perusteella oikean view-komponentin.

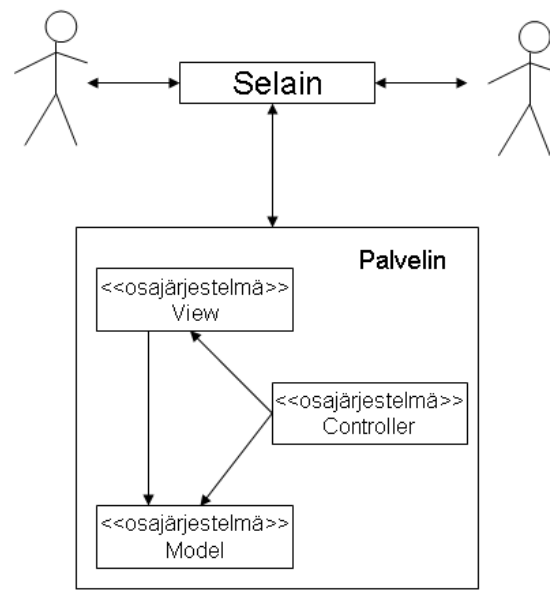
Kuva 1 esittää järjestelmän muodostumisen osajärjestelmistä sekä käyttäjistä.

3.2 Komponentit

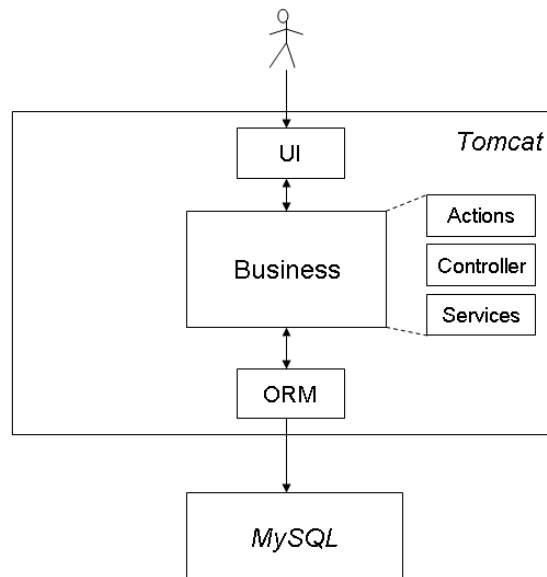
Tässä esitetään osajärjestelmiin liittyviä komponentteja. Kuva 2 esittää alla mainittujen komponenttien sijoittumisen järjestelmään sekä niiden suhteen toisiinsa. Kuvasta ilmenee kuinka UI-komponentti tarjoaa liittymän käyttäjälle. ORM-komponentti hoitaa olioien tallennuksen tietokantaan sekä kaikki niihin liittyvät tietokantaoperaatiot. Business toimii näiden välissä toteuttaen Struts-sovelluskehiksen avulla MVC-arkkitehtuurityylin. Kaikki komponentit vaativat toimiakseen Tomcat-komponentin jonka sisälle ne sijaitsevat.

3.2.1 UI (User Interface)

UI-komponentti toteutetaan käyttäen JavaServer Pages (JSP) -tekniikkaa, joka yhdistää XHTML:ää ja Java-koodia samaan tiedostoon dynaamisten näyttö sivujen aikaan saamiseksi. JSP-tiedoston koodiosista generoidaan Servlet-tiedosto, joka ajetaan sovelluspal-



Kuva 1: Osajärjestelmät



Kuva 2: Komponentit

velimella. JSP-sivulla käytetään Struts:in tarjoamia tajeja, joiden avulla päästään käsiksi Business-komponentin tarjoamiin palveluihin.

UI komponentti toteuttaa MVC-tyylin View-osan.

3.2.2 Business

Business-komponentti toteuttaa sovelluksen kaiken toimintalogiikan. Komponentti perustuu MVC-arkkitehtuurityyliin, jonka toteuttamiseen käytetään Struts-sovelluskehystä.

Struts-ohjelmistokehys on Java Servlet -tekniikkaan perustuvien www-sovellusten kehittämiseen tarkoitettu ohjelmistokehys. Struts-sovelluskehysten tärkein ominaisuus on se, että ohjelmoijan ei tarvitse kantaa huolta MVC-arkkitehtuurityylin säilymisestä, sillä kehysten menettelyitä seuraamalla sovellukset säilyttävät automaattisesti MVC-tyylin.

Käytännössä Business-komponentti jakautuu Actions, Controller ja Services -alikomponentteihin. Actions-komponentti tarjoaa UI-komponentille rajapinnan sovelluksen toimintalogiikkaan. Controller toimii käyttäjän ja muiden komponenttien välissä. Se ohjaa käyttäjän pyynnöt Action-komponentille, joka puolestaan käyttää Services-komponenttia. Services-komponentti toteuttaa erilaisia palveluita, esimerkkinä Tietokanta-operaatiot sekä Sessionhallinta. Controller myös ohjaa UI-komponenttia, jotta käyttäjälle näytetään oikea sivu vastauksena.

Business-komponentti toteuttaa MVC-tyylin Controller-osan sekä yhdessä ORM-komponentin kanssa Model-osan.

3.2.3 ORM (Object-relational mapping)

ORM-komponentti toimii MySQL- ja Business-komponenttien välissä. Sen avulla voidaan tallentaa olioita relaatiotietokantaan, mikä helpottaa sovelluksen toteuttamista ja vähentää tarvittavien SQL-lauseiden määrää.

ORM perustuu suurelta osin Hibernateen, joka on työkalu olio-relaatio-vastaavuuden toteuttamiseen. Hibernaten avulla olioita voidaan tallentaa tietokantaan ilman SQL-komentojen kirjoittamista, aivan kuin tietokanta olisi relaatiopohjaisuuden sijasta oliopohjainen. Hibernate lisää myös ylläpidettävyyttä mahdollistaen tietokannan vaihtamisen toiseen järjestelmän sovelluslogiikkaa muuttamatta.

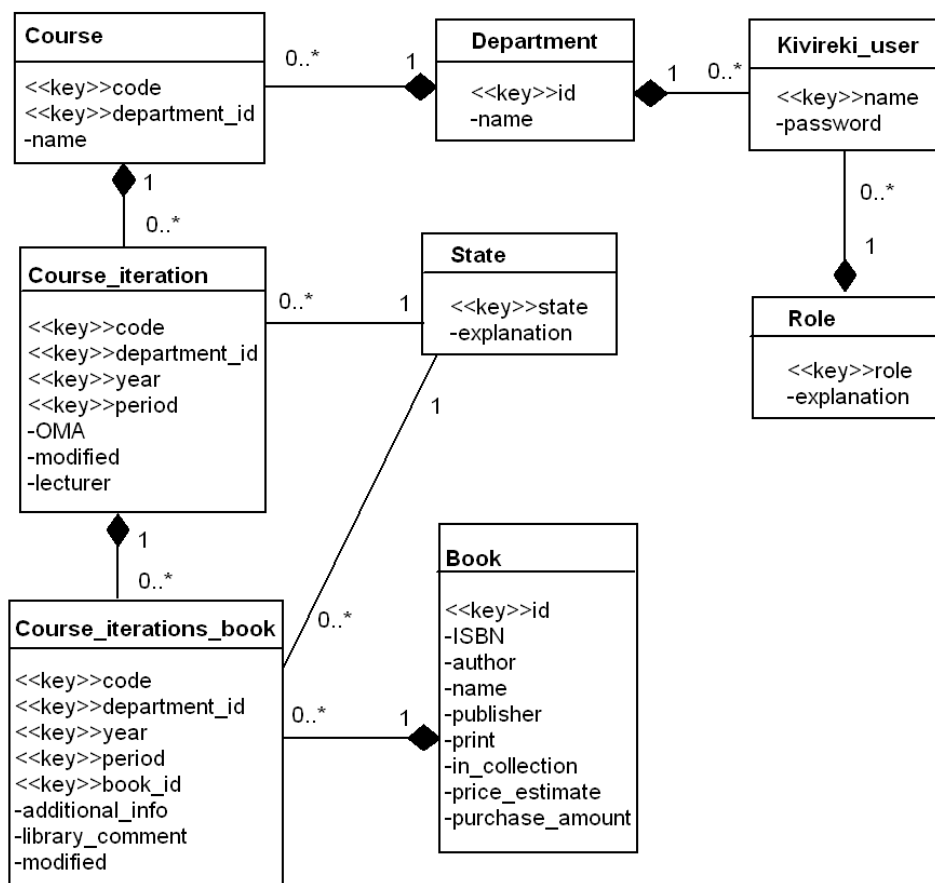
ORM-komponentti toteuttaa MVC-tyylin Model-osaa Business-komponentin kanssa.

3.2.4 MySQL

MySQL on projektin monisäikeinen, monen käyttäjän tietokantapalvelin, johon säilötään järjestelmän tarvitsemat tiedot Hibernaten avulla.

3.2.5 Tomcat

Tomcat on projektin sovelluspalvelin. Se toteuttaa Sun Microsystemsin Java Servlet- ja JSP-spesifikaatiot, jotka Struts vaatii toimiakseen. Tomcat välittää asiakasohjelman sivunhakupyynnöt Struts:lle mahdollistaen Java-koodin suorituksen yhteistyössä web-palvelinympäristön kanssa.



Kuva 3: Tietokannan rakenne

4 Tietokanta

Tietokannan sisältö ja rakenne on esitelty kuvassa 3. «key» -merkityt attribuutit tarkoittavat taulun avainattributteja. Tietokannan suunnittelussa on otettu huomioon järjestelmän mahdollinen laajentuminen koko yliopiston käyttöön luomalla department-taulu, jonka avulla voidaan kurssit ja kurssi-iteraatiot yhdistää tiettyyn laitokseen. Tietokannan taulut käyttävät tietokantamoottorina InnoDB-tietokantamoottoria, joka tuo hyödyllistä lisätoiminnallisuutta tietokantaoperaatioihin. InnoDB tukee mm. viiteavaimia ja niihin liittyviä viiteavainrajoitteita.

4.1 Käyttäjä

Kivireki_user-taulu sisältää käyttäjien tiedot. Role tarkoittaa käyttäjäryhmää, ja department_id on laitos, jossa käyttäjä työskentelee (kirjasto on myös laitos).

```
CREATE TABLE kivireki_user(
```

```

name          VARCHAR(100) NOT NULL,
password      VARCHAR(100) NOT NULL,
role          INT UNSIGNED NOT NULL,
department_id INT UNSIGNED NOT NULL,

PRIMARY KEY kivireki_user_key (name),
FOREIGN KEY (department_id) REFERENCES department (id),
FOREIGN KEY (role) REFERENCES role (role)
) ENGINE=InnoDB;

```

4.2 Kirja

Book-tauluun talletetaan kurssikirjojen tiedot. Kirjat yksilöidään automaattisella id-numerolla. Pakollisia tietoja kirjoista ovat kirjoittaja, kirjan nimi, kustantaja sekä painosnumero (`print`). Lisäksi kirjasta voidaan tallettaa tieto onko kirjaa kirjaston kokoelmassa (`in_collection`), hinta-arvio sekä tilattava määrä (`purchase_amount`). Kirjoittaja syötetään muodossa “sukunimi, etunimi” (ilman lainausmerkkejä). Jos kirjalla on useampia kirjoittajia, erotetaan ne toisistaan puolipisteellä. ISBN-numero syötetään ilman väliviivoja, jos niitä on useampia, ne erotetaan toisistaan pilkulla.

```

CREATE TABLE book(
  id          INT UNSIGNED NOT NULL AUTO_INCREMENT,
  ISBN       VARCHAR(100) NOT NULL,
  author     VARCHAR(100) NOT NULL,
  name       VARCHAR(100) NOT NULL,
  publisher  VARCHAR(100) NOT NULL,
  print      INT NOT NULL,
  in_collection INT,
  price_estimate INT,
  purchase_amount INT,

  PRIMARY KEY book_key (id, author, name, publisher, print)
) ENGINE=InnoDB;

```

4.3 Kurssi

Course-taulussa eri kurssit yksilöidään kurssikoodin (`code`) ja laitoksen (`department_id`) avulla, sillä eri laitoksilla saattaa olla kursseja samalla koodilla. Kurssiin voidaan liittää myös nimi. Kurssikoodi tulee syöttää ilman väliviivoja.

```

CREATE TABLE course(
  code        INT UNSIGNED NOT NULL,
  name       VARCHAR(100) NOT NULL,
  department_id INT UNSIGNED NOT NULL,

  PRIMARY KEY course_key (code, department_id),
  FOREIGN KEY (department_id) REFERENCES department (id)
) ENGINE=InnoDB;

```

4.4 Kurssikerta

Course_iteration-taulu kuvaa yksittäistä kurssikertaa. Kurssikerta yksilöidään koodin, laitoksen, vuoden ja periodin avulla. Lisäksi kurssikerran tietoihin kuuluu osallis-

tujamääräarvio (OMA), tila, luennoija sekä muokkauspäivä. Mikäli luennoijia on useita, syötetään kurssin vastuuhenkilön nimi. Luennoijan nimi syötetään muodossa “sukunimi, etunimi” (ilman lainausmerkkejä).

```
CREATE TABLE course_iteration(
  code          INT UNSIGNED NOT NULL,
  department_id INT UNSIGNED NOT NULL,
  year          INT UNSIGNED NOT NULL,
  period        INT UNSIGNED NOT NULL,
  OMA           INT,
  state         INT UNSIGNED NOT NULL,
  modified      DATE,
  lecturer      VARCHAR(100) NOT NULL,

  PRIMARY KEY course_iteration_key (code, department_id, year, period),
  FOREIGN KEY (code, department_id) REFERENCES course (code, department_id),
  FOREIGN KEY (state) REFERENCES state (state)
) ENGINE=InnoDB;
```

4.5 Kurssikerran kirja

Taulu liittyy yhden kirjan tiettyyn kurssikertaan. Yhdellä kurssikerralla voi olla useampia kirjoja. Kurssikerran kirja yksilöidään kuten kurssikerta, lisäksi kirjasta kirjasta id (book_id). Lisäksi on kenttä luennoijan antamille lisätiedoille (additional_info), kirjaston kommenttikenttä, kirjailmoituksen tila sekä muokkauspäivä.

```
CREATE TABLE course_iterations_book(
  code          INT UNSIGNED NOT NULL,
  department_id INT UNSIGNED NOT NULL,
  book_id       INT UNSIGNED NOT NULL,
  year          INT UNSIGNED NOT NULL,
  period        INT UNSIGNED NOT NULL,
  additional_info VARCHAR(100),
  library_comment VARCHAR(100),
  state         INT UNSIGNED NOT NULL,
  modified      DATE,

  PRIMARY KEY course_iterations_book_key (code, department_id, year, period, book_id),
  FOREIGN KEY (book_id) REFERENCES book (ID),
  FOREIGN KEY (state) REFERENCES state (state),
  FOREIGN KEY (code, department_id, year, period)
  REFERENCES course_iteration (code, department_id, year, period)
) ENGINE=InnoDB;
```

4.6 Laitos

Department-tauluun tallennetaan eri laitosten nimet. Laitokset yksilöidään automaattisella id-numerolla (järjestelmän mahdollinen laajennettavuus on otettu huomioon).

```
CREATE TABLE department (
  id          INT UNSIGNED NOT NULL AUTO_INCREMENT,
  name        VARCHAR(100) NOT NULL,

  PRIMARY KEY department_key (id)
) ENGINE=InnoDB;
```

4.7 Tila

State-taulu toimii selittävänä tauluna. Siitä löytyy ohjelmoijan avuksi selitteet tilojen numeroille sanallisessa muodossa. Käytetään kurssi-iteraatioiden ja kurssi-iteraatioiden kirjojen tilatietoihin.

```
CREATE TABLE state (  
    state          INT UNSIGNED NOT NULL AUTO_INCREMENT,  
    explanation    VARCHAR(100) NOT NULL,  
  
    PRIMARY KEY state_key (state)  
) ENGINE=InnoDB;
```

4.8 Rooli

Role-taulu toimii selittävänä tauluna. Siitä löytyy ohjelmoijan avuksi selitteet eri roolien numeroille sanallisessa muodossa. Käytetään käyttäjäryhmien identifioimiseen.

```
CREATE TABLE role(  
    role          INT UNSIGNED NOT NULL,  
    explanation    VARCHAR(100) NOT NULL,  
  
    PRIMARY KEY role_key (role)  
) ENGINE=InnoDB;
```

5 Tiedostot

Järjestelmän tiedostot voidaan jakaa kuuteen kategoriaan. Järjestelmän Java-luokat ja kaantuvat Strutsin arkkitehtuurin mukaisesti Business-, Model- ja Action-osiin. Lisäksi järjestelmä sisältää edellä mainittuihin Java-luokkiin liittyviä kartoitustiedostoja. Käyttöliittymän toteutus koostuu JSP-sivuista ja niihin liittyvistä CSS-tiedostoista.

5.1 Business-luokat

Business-luokat sisältävät Model-luokkien kanssa MVC-mallin mukaisen toteutuksen Model-osion toiminnallisuuden. Business-luokat hoitavat kaiken tietokantaoperaatioihin liittyvän toiminnallisuuden Model-luokkien avulla. Konkreettiset tietokantaoperaatiot, kuten SQL- tai HQL-kieliset tietokantakyselyt, tehdään Business-luokan EventService-rajapinnan toteuttavassa luokassa EventServiceImpl. Hibernate generoi HQL-kyselyjen perusteella tietokantaa varten oikeat SQL-kyselyt ja suorittaa ne tallentaen mahdolliset tulokset Model-luokissa toteutettuihin persistenssiolioihin. SessionService-rajapinnan toteuttava SessionServiceImpl-luokka sisältää tarvittavan toiminnallisuuden sessioiden hallintaan, eli tiedon kirjoittamiseen sessioon, tiedon lukemiseen sessiosta ja session aikakatkaisuun.

5.2 Model-luokat

Model-luokat sisältävät business-luokkien kanssa MVC-mallin mukaisen toteutuksen Model-osion toiminnallisuuden. Ne ovat yksinkertaisia persistenssiluokkia, jotka vastaavat muutujakentiltään tietokannan tauluja ja sisältävät get- ja set-metodit arvojen saamiseen ja muuttamiseen. Hibernate tallentaa olioihin tietokantakyselyistä saamansa tulokset ja käyttää niitä tallentaessaan uusia rivejä tietokantaan sekä erilaisiin hakutoimenpiteisiin.

5.3 Action-luokat

Action-luokat sisältävät MVC-mallin mukaisen toteutuksen Model-osion toiminnallisuutta. Luokat ovat järjestelmässä osa Business-komponenttia. Asiakasohjelman pyynnöt saavat Struts-sovelluskehikössä aikaan kutsun action-luokkaan siten kuin on määritelty kartoitustiedostossa. Action-luokissa käsitellään service-luokilta saatu tieto JSP-sivujen tarvitsemaan muotoon. Action-luokat tarjoavat myös rajapinnan JSP-sivuille tiedon saamiseen ja päinvastoin; JSP-sivujen tarjoamien tietojen lukemiseen action-luokassa.

5.4 Kartoitustiedostot (konfiguraatitiedostot)

Kartoittavat Struts-sovelluskehiksen ja Action-luokkien sekä Hibernate-palvelun ja tietokannan välisen kuulun. Kartoitustiedostot ovat XML-kielillä kirjoitettuja. Ne kertovat Strutsille, mikä action-luokka vastaa mitään asiakasohjelman sivunhakupyynnöä (jos vastaavuutta ylipäätään on). Lisäksi kartoitustiedostot kertovat, mikä JSP-sivu on näytettävä

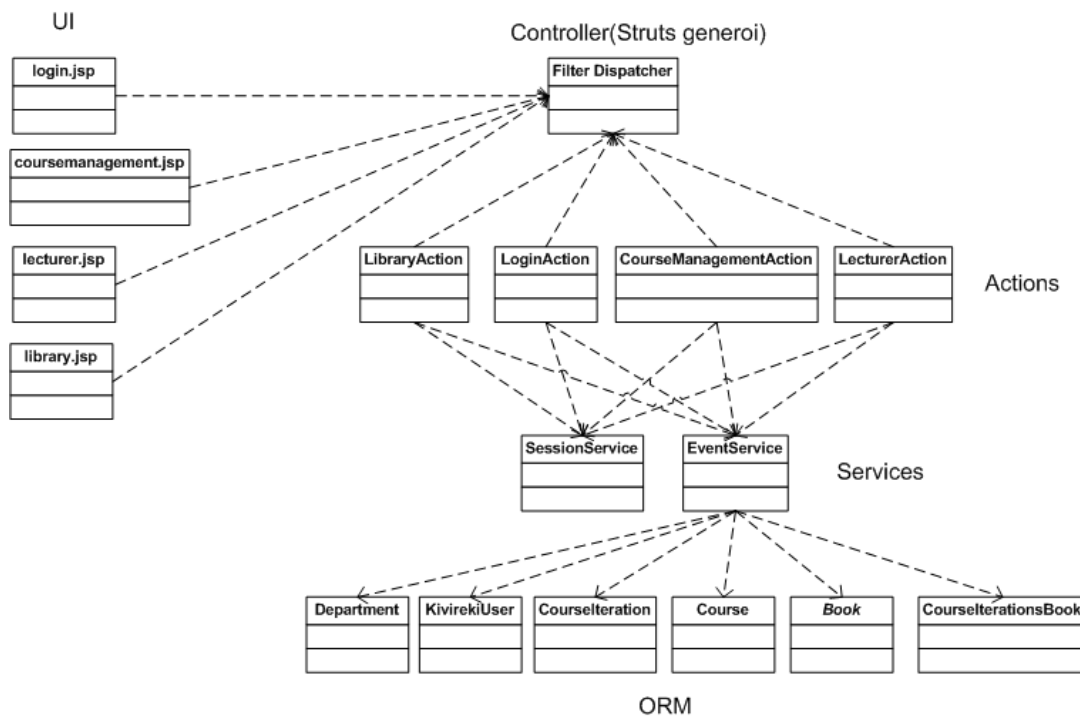
action-luokan suorituksen jälkeen riippuen action-luokan palauttamasta String-muotoisesta paluuarvosta. Hibernate puolestaan käyttää kartoitustiedostoja määrittääkseen mikä persistenssiluokka vastaa mitäkin tietokannan taulua ja mikä persistenssiluokan kenttä vastaa mitäkin taulun saraketta. Hibernate osaa tämän jälkeen tehdä kyselyjä tauluihin ja alustaa persistenssiluokkien kentät oikein.

5.5 JSP-sivut

JSP-tiedostot sisältävät MVC-mallin mukaisen toteutuksen View-osion toiminnallisuuden. Ne sisältävät XHTML-kieltä ja Java-koodia. Struts määrää mikä JSP-sivu näytetään asiakkaalle action-luokan suorituksen jälkeen siten kuin sen kartoitustiedostossa on määritetty. Järjestelmän dynaamiset näyttösivut ovat JSP-kielisiä. Ne saavat Action-luokilta tarvitsemansa tiedon ja tarjoavat action-luokille tietoa GET- ja POST-parametrien muodossa.

5.6 CSS-tiedostot

CSS-tyylitiedostot sisältävät käyttäjälle näytettävien web-sivujen ulkoasun määrittelyt. Käytettäviin tyylitiedostoihin viitataan JSP-sivuista ja ne määräävät lopullisen asiakasohjelmalla näkyvän näyttösivun ulkoasun.



Kuva 4: Luokkakaavio

6 Luokkien kuvaukset

6.1 Luokkakaavio

Järjestelmän luokkakaavio on kuvassa 4.

6.2 JSP-sivut

Jokaisen JSP-sivun HTML-perusrakenne on seuraava:

- html (HTML-dokumentti)
 - head (otsaketiedot)
 - * title (sivun otsikko)
 - body (itse sivu)
 - * div id="header" (sivun otsakeosio, "ylätunniste")
 - * div id="content" (sivun pääasiallinen sisältö)
 - * div id="footer" (mahdolliset loppuosat, "alätunniste")

Jokaisen sortable-luokan taulukon rakenne on seuraava:

- `thead` (otsake-osio)
 - `tr class="titles"` (sarakkeiden otsikkorivi)
 - `tr class="filters"` (hakukenttärivi)
- `tbody` (sisältöosio)
- `tfoot` (mahdollinen yhteenveto-osio)

Järjestelmä sisältää seuraavat JSP-sivut:

6.2.1 `login.jsp`

Kuvaus: Sisäänkirjautumissivu, jossa käyttäjä syöttää käyttäjätunnuksen ja salasanan kirjautuakseen sisään.

Rakenne:

- `form id="login"` (kirjautumislaatikko)

Sisäänkirjautumissivu näkyy kuvassa 5.



The image shows a login form with the following structure:

```

<h3>Kirjautuminen</h3>
<input type="text" value="Tunnus" />
<input type="password" value="Salasana" />
<input type="button" value="Sisään" />

```

Kuva 5: kirjautuminen

6.2.2 `coursemanagement.jsp`

Kuvaus: Kurssihallinnon näkymä, jossa näytetään ja muokataan kursseja ja kurssi-iteraatioita ja lähetetään ilmoitus kirjastolle muuttuneesta tilanteesta.

Rakenne:

- `form id="yearSelector"` (muokattavan vuoden valitsin [1])
- `table id="currentCourses" class="sortable"` (muokattavan vuoden kurssitaulukko)
- `form id="newCourse"` (uuden kurssin lisääminen [7])
- `form id="sendNotice"` (ilmoituksen lähettäminen [8])

Kurssit

1. vuoden valitsin 2. sarakkeiden otsikot 3. suodattava haku

Muokattava vuosi:

Periodi(t)	Koodi	Kurssi	Luennoitsija	Tekijä	Kirja	Julkaisija	Painos	Lisätiedot	OK	OMA
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	1561328	Ohjelmoinnin perusteet	Wikla Arto	Wikla Arto	Ohjelmoinnin perusteet... Java-kielillä	Otadata	3	Kirja on verkossa, ei pakollinen.	-	112
	4324256	Java-ohjelmointi	Wikla Arto	Wikla Arto	Ohjelmoinnin perusteet... Java-kielillä	Otadata	3	Kirja on verkossa, ei pakollinen.	-	30
<input checked="" type="checkbox"/>	4891357	Johdatus tietojenkäsittelytieteeseen	Niklander Tiina						<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	1651168	Rinnakkaisohjelmointi	Kerola Teemu	Ben-Ari	Principles of Concurrent Programming	Addison Wesley	2	Välttämätön hankinta.	-	
				Andrews G.P.	Foundation of Distributed Programming	Addison Wesley	6	Oheislukemistoa.	-	55
				Stallings	Operating Systems	Prentice Hall	5	Sisältää yhden kurssiin kuuluvan luvun, jota ei ole Ben-Arin kirjassa.	-	
<input checked="" type="checkbox"/>	4564123	Peliohjelmointi (C++)	Otto Nurmi						<input type="checkbox"/>	

4. kurssin tiedot 5. kurssikerran kirjat

6. kurssikerran muut tiedot

Uusi kurssi: Koodi: Nimi: 7. uuden kurssin lisääminen

8. "kirjastolle ilmoitus" -nappi

Vanhat kurssit 9. taulukko aiempien vuosien kurssikerroista

Vuosi	Periodi(t)	Koodi	Kurssi	Luennoitsija	Tekijä	Kirja	Julkaisija	Painos	Lisätiedot	
2007	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		4891357	Johdatus tietojenkäsittelytieteeseen	Niklander Tiina						Kopioi
2007	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		1561328	Ohjelmoinnin perusteet	Wikla Arto	Wikla Arto	Ohjelmoinnin perusteet... Java-kielillä	Otadata	3	Kirja on verkossa, ei pakollinen.	Kopioi
2007	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		4324256	Java-ohjelmointi		Wikla Arto	Ohjelmoinnin perusteet... Java-kielillä	Otadata	3	Kirja on verkossa, ei pakollinen.	Kopioi
2007	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		1561328	Ohjelmoinnin perusteet	Wikla Arto	Wikla Arto	Ohjelmoinnin perusteet... Java-kielillä	Otadata	3	Kirja on verkossa, ei pakollinen.	Kopioi
					Ben-Ari	Principles of Concurrent Programming	Addison Wesley	2	Välttämätön hankinta.	

Kuva 6: kurssihallinnon käyttöliittymä

- table id="oldCourses" (aiempien vuosien kurssitaulukko [9])

Kurssihallinnon sivu näkyy kuvassa 6.

6.2.3 lecturer.jsp

Kuvaus: Luennoitsijoiden näkymä, jossa näytetään ja muokataan kurssikirjatietoja sekä lisätään, muokataan ja poistetaan kirjoja järjestelmästä.

Rakenne:

- form id="yearSelector" (muokattavan vuoden valitsin)
- table id="currentCourses" class="sortable" (muokattavan vuoden kurssitaulukko [1])
- table id="booksLecturer" (kirjataulukko [2])

Kurssit

1. muokattavien kurssien taulukko kuten kurssihallinnolla

Muokattava vuosi: - 2008 +

Periodi(t)	Koodi	Kurssi	Luennoitsija	Tekijä	Kirja	Julkaisija	Painos	Tiätiedot	OK	OMA
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2. järjestelmässä jo olevat kirjat

Lisää kirja

Tekijä	Kirja	Painos	Julkaisija	Lisää kurssiin
Wikla Arto	Ohjelmoinnin perusteet... Java-kielillä	2	Otadata	Valitse kurssi
Wikla Arto	Ohjelmoinnin perusteet... Java-kielillä	3	Otadata	Valitse kurssi
Ben-Ari	Principles of Concurrent Programming	3	Addison-Wesley	Valitse kurssi
Stallings	Operating Systems	5	Prentice Hall	Valitse kurssi
Stallings	Operating Systems	6	Prentice Hall	Valitse kurssi
Andrews G.P.	Foundation of Distributed Programming	6	Addison Wesley	Valitse kurssi
Petroskoi	Principles of Multithreaded Programming	1	Prentice Hall	Valitse kurssi

4. kirjan tiedot

5. kirjan lisääminen kurssiin

Uusi kirja	Valitse kurssi
<input type="text"/>	Valitse kurssi

3. uuden kirjan lisääminen

Kuva 7: luennoitsijoiden käyttöliittymä

- form id="newBook" (uuden kirjan lisääminen [3])

Luennoitsijoiden sivu näkyy kuvassa 7.

6.2.4 library.jsp

Kuvaus: Kirjaston näkymä, jossa näytetään kirjatilaustiedot ja käyttäjät voivat lisätä tietoja kirjojen hinnasta ja tilasta sekä muokata tilausten käsittelystatusta.

Rakenne:

- form id="departmentSelector" (laitosvalitsin [1])
- form id="viewOptions" (näkymän rajoittaminen [2])
- table id="booksLibrary" class="sortable" (kirjataulukko)

Kirjaston sivu näkyy kuvassa 8.

Näytä vain uusien kurssikerta
 Näytä vain kirjat, joita hyllyssä

2. näkymän rajoittaminen

1. laitosvalitsin 3. sarakkeiden otsikot 4. suodattava haku

Kurssikirjat

Tekijä	Nimeke	Julkaisija	Painos	Kurssi	Luennointsija	Laitos	Lukuvuosi	Periodi	OMA	Lisätiedot	Muutos	Kokoel.	Hank.	ä	Yht. Käs.	Kommentti
Wikla Arto	Ohjelmoinnin perusteet, Java-kielellä	Otadata	3	Java ohjelmointi	Wikla, Arto	TKTL	2008-2009	1	100	Kirja on verkossa, ei pakollinen.	uusi	15	0	42	0	
Ben-Ari	Principles of Concurrent Programming	Addison Wesley	2	Rinnakkaisohjelmointi	Pasanen, Tomi	TKTL	2008-2009	1	35	Kirja on verkossa, ei pakollinen.	uusi	0	0	0	690	
Andrews G.F.	Foundation of Distributed Programming	Addison Wesley	6	Rinnakkaisohjelmointi	Kerola, Teemu	TKTL	2008-2009	1	85	Välttämätön bankia.	uusi	0	0	0	0	
Stallings	Operating Systems	Prentice Hall	5	Rinnakkaisohjelmointi	Kerola, Teemu	TKTL	2008-2009	1	85	Oheislukemistoa.	uusi	0	0	0	0	
Koskimies, Kai	Ohjelmistoarkkitehtuurit	Talentum	1	Ohjelmistoarkkitehtuurit	Viljamaa, Jukka	TKTL	2008-2009	1	51 (30)	Sisältää yhden kurssin kuuluvan luvun, jota ei ole Ben-Arin kirjassa.	muutos	3	0	49	49	
Sommerville	Software Engineering	Addison-Wesley	9	Ohjelmistojen suunnittely	Viljamaa, Jukka	TKTL	2008-2009	2	50	ei pakollinen, opetusmoniste myös saatavilla	peruttu	0	10	119	1190	
				Ohjelmistotuotanto	Taina, Juha	TKTL	2008-2009	1	65							
(kurssikirjoja ei vielä määritetty)				Laskennan mallit	Karvi, Timo	TKTL	2008-2009	1	75							
Alan Webb	Software Engineering	Addison-Wesley	2	Ohjelmistojen vaatimusmäärittely	Taina, Juha	TKTL	2008-2009	1	22	Kirja on erittäin tarpeellinen.		0	2	55	110	✓
Petroski	Computing Analysis	Prentice Hall	1	Laskennan vaativuus	Kivinen, Jyrki	TKTL	2008-2009	1	28	Normaali kurssikirja.		2	0	0	0	✓
James Brown	User Interfaces	Prentice Hall	11	Käyttöliittymät	Luontola, Esko	TKTL	2008-2009	1	123			15	0	88	0	✓
Aristoteles	Principles of Logic	Greece Publishing	42	Logiikkaohjelmointi	Kivinen, Jyrki	TKTL	2008-2009	1	5	lisälukemistoa		2	0	6,5	0	✓
				Logiikka IV	Hyttinen, Tapani	Mat	2008-2009	1	20	Välttämätön!						

5. kirjan tiedot 6. kirjoihin liittyvien kurssien tiedot 7. hankinta-/käsittelytiedot

Kuva 8: kirjaston käyttöliittymä

6.3 CSS-tyylitiedostot

Kaikki tyylimäärittelyt tehdään juurihakemistossa olevaan `main.css`-tiedostoon.

6.4 Java-luokat

Java-luokkien yleiskuvaukset sekä rajapinnat ovat javadoc-muodossa liitteenä.

A Java-luokat