

HELSINGIN YLIOPISTO

# Lahjalistasovellus

---

Tietokantasovellus, Syksy 2014

**Tomi Kasurinen**

## 1 Johdanto

Lahjalistapalveluun ylläpitäjän voivat lisätä toivomiansa lahjoja, esim. häihin tai muihin juhliin liittyen. Vieraat saavat linkin sivulle, josta he pystyvät selaamaan lahjoja. Lahjan valittuaan vieraat voivat varata sen täyttämällä tietonsa. Jo ennestään varanneet vieraat voivat varata lahjan käyttämällä sähköpostiosoitettaan. Ylläpitäjät voivat antaa esimerkkipaikan mistä lahjan voi ostaa. Kun lahja on varattu, se merkitään myös lahjalistaan varatuksi muille vieraille. Jotkut lahjat voi myös varata/ostaa monta kertaa, jos ylläpitäjä on niin määritellyt.

Ylläpitäjät voivat ylläpitäjätunnuksilla lisätä ja poistaa lahjaehdotuksia sekä muokata varauksia.

Ohjelman sovelluslogiikka toteutetaan Java-kielellä ja tietokantana käytetään SQL-tietokantaa ja PostgreSQL-tietokantajärjestelmää. Ohjelma toteutetaan Helsingin yliopiston Tietojenkäsittelytieteen laitoksen users-palvelimella, josta löytyy tarvittava Java-tuki.

Ohjelman toimimiseksi selaimen on tuettava javascript-kieltä. Ohjelma on myös toteutettu PostgreSQL-tietokantajärjestelmää käyttäen, ja muita tietokantajärjestelmiä varten ylläpitäjä joutuu todennäköisesti modifiomaan ohjelmaa.

## **2 Yleiskuva järjestelmästä**

### **2.1 Käyttäjärühmät**

#### **Ylläpitäjä**

Ylläpitäjä on sivuston ylläpitäjä joka pystyy ylläpitäjätunnusten kautta lisäämään, muokkaamaan ja poistamaan lahjaehdotuksia ja -varauksia. Ylläpitäjänä toimii yleensä juhlien järjestäjä.

#### **Vieras**

Vieras on juhliin osallistuva / kutsuttava henkilö, joka on saanut ylläpitäjältä linkin palveluun. Tämän avulla vieras pystyy selaamaan ja varaamaan lahjaehdotuksia.

### **2.2 Käyttötapaukset**

#### **2.2.1 Ylläpitäjän käyttötapaukset**

##### **Lahjaehdotuksen lisääminen**

Ylläpitäjä voi lisätä lahjaehdotuksia vieraille selattavaksi ja varattaviksi. Ylläpitäjä myös mm. määrittää kuinka monta kertaa lisätyn lahjaehdotuksen voi varata.

##### **Lahjaehdotuksen muokkaaminen/poistaminen**

Ylläpitäjä pystyy myös muokkaamaan ja poistamaan olemassa olevia lahjaehdotuksia.

## **Lahjavarausten muokkaaminen/poistaminen**

Ylläpitäjä pystyy myös muokkaamaan ja poistamaan olemassa olevia lahjaehdotuksia, esim. Vieraan pyynnöstä.

**Muita käyttötapauksia:** Kirjautuminen, lahjaehdotusten selaus ja lahjavarausten selaus

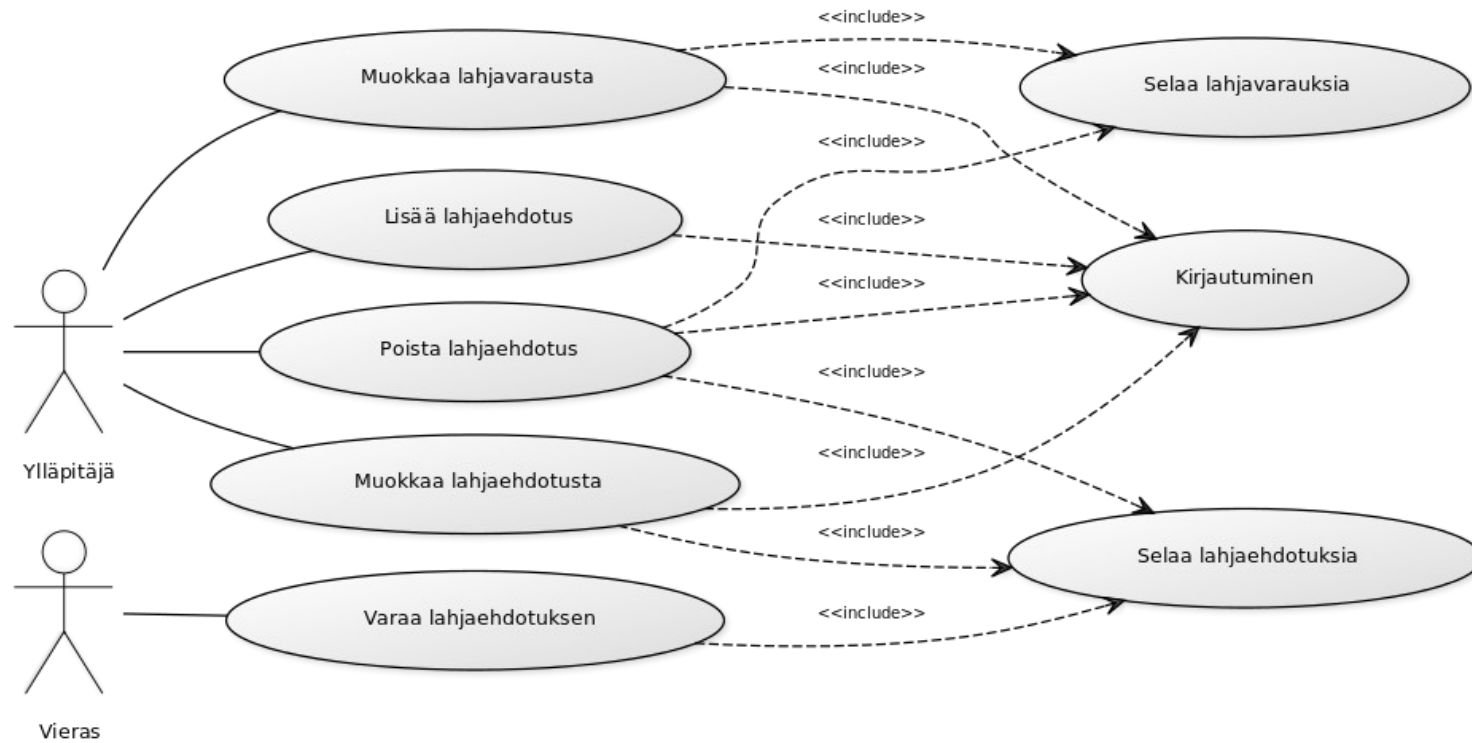
### **2.2.2 Vieraan käyttötapaukset**

#### **Lahjaehdotuksen varaus**

Vieraat voivat varata lahjaehdotuksen selattuaan niitä ja löydyttyään sopivan, jos sitä ei ole vielä varattu ylläpitäjän määrittämää maksimäärää.

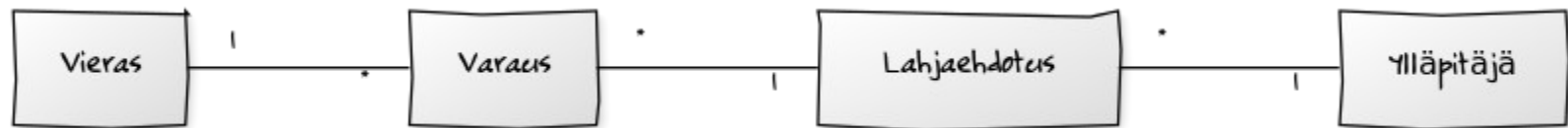
**Muita käyttötapauksia:** Kirjautuminen, lahjaehdotusten selaus.

### 2.2.3 Käyttötapauskaavio



### 3 Järjestelmän tietosisältö

#### 3.1 Käsitetason tietokantamalli



#### 3.2 Tietokohteiden määrittely

##### 3.2.1 Lahjaehdotus

Atribuutti	Arvojoukko	Kuvailu
id	integer	Itsestään täydentyvä id lahjaehdotuksille
nimi	Varchar(30)	Lahjalle annettu nimi
ostoOsoite	Varchar(50)	Esimerkki osto-URL lahjalle, voi olla null.
lisääjä	Varchar(30)	Viite Ylläpitäjä taulun käyttäjätunnukseen, ilmaisee lahjan lisääjän
maxVaraukset	int	Ilmaisee kuinka monta kertaa lahjan voi maksimissaan varata

##### 3.2.2 Vieras

Atribuutti	Arvojoukko	Kuvailu
id	integer	Itsenstään täydentyvä id-tunniste vieraille
nimi	varchar	Vieraan nimi
email	varchar	Vieraan email, käytetään myös uudelleenvarauksessa tunnisteena
puhno	varchar	Vieraana puhelinnumero

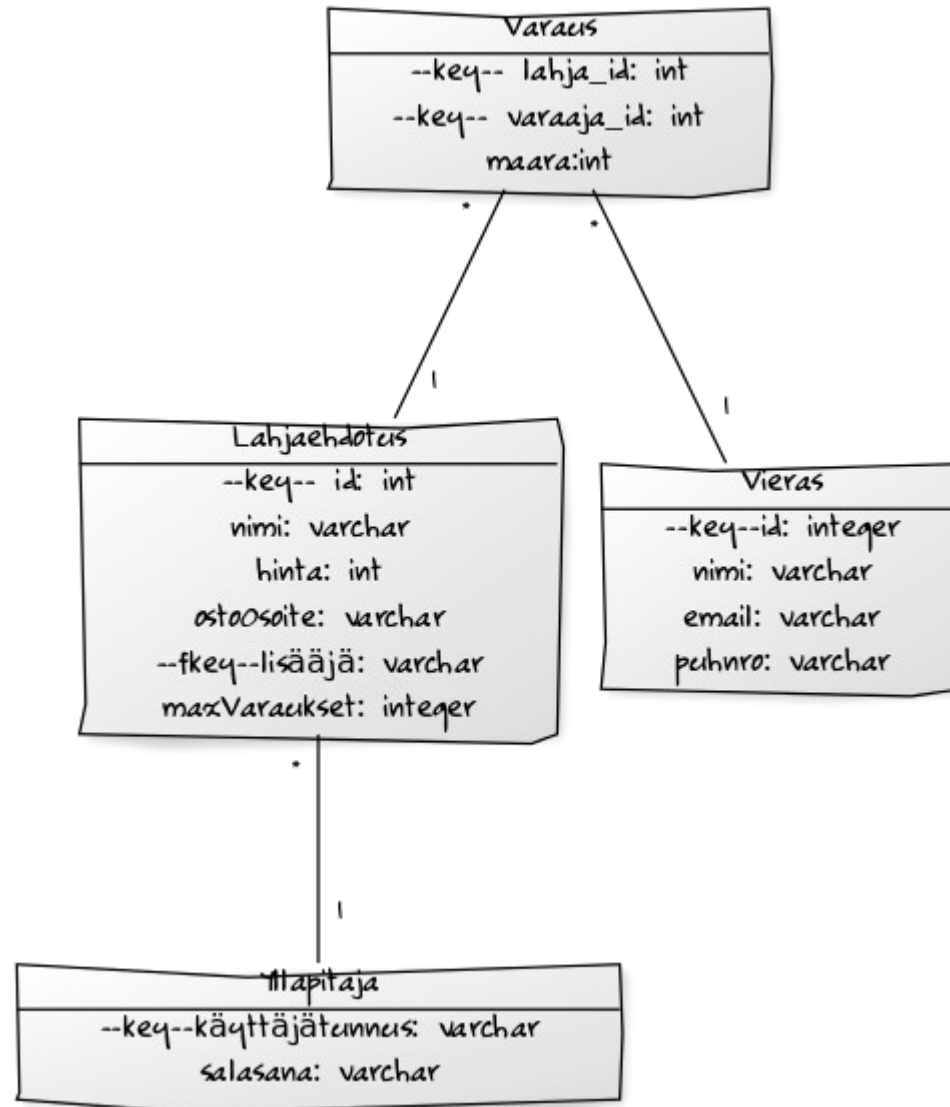
### 3.2.3 Varaus

Atribuutti	Arvojoukko	Kuvailu
lahja_id	integer	Varaukseen liittyvän lahjan id-tunniste
varaaja_id	integer	Varauksen tehneen vieraan id-tunniste

### 3.2.4 Ylläpitäjä

Atribuutti	Arvojoukko	Kuvailu
kayttajatunnus	varchar	Ylläpitäjän käyttäjätunnus, toimii myös primary key:na
salasana	varchar	Ylläpitäjän salasana kirjautumiseen

#### 4 Relaatiotietokantakaavio





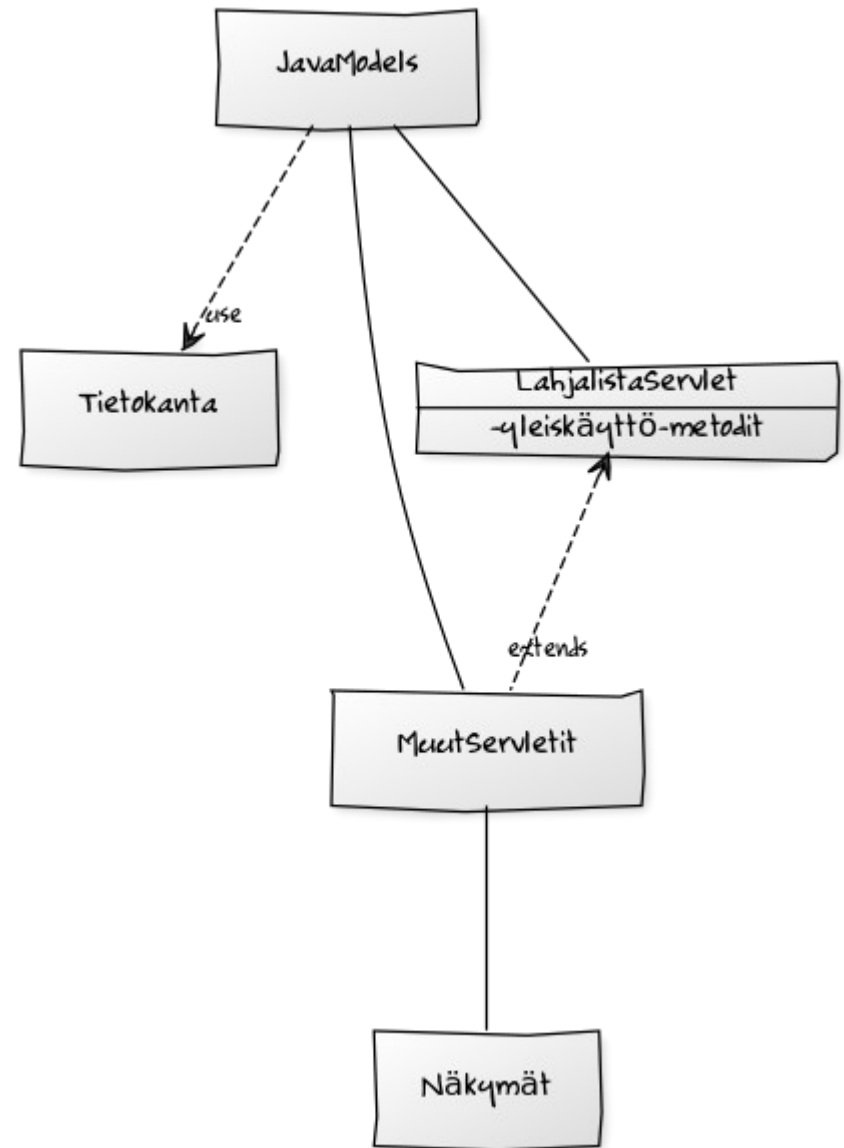
## 5 Järjestelmän yleisrakenne

Tietokantasovellus on toteutettu hydyntäen MVC-mallia.

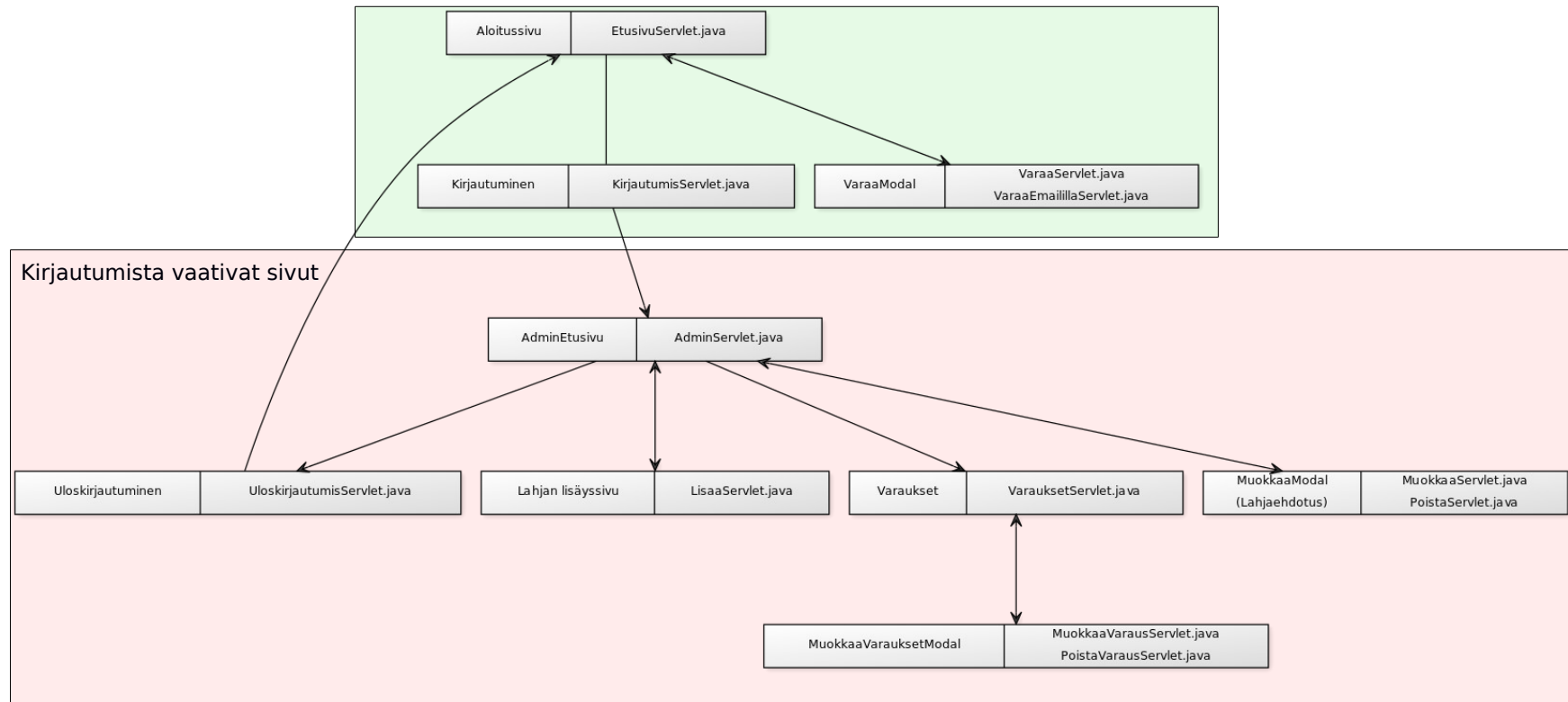
Mallit on sijoitettu kansioon src/java/Models. Javan malliluokkia on yksi joka relaatiotietokantaa varten.

Kontrollereina toimivat Java-servletit, ja ne on sijoitettu kansioon src/java/Servlets. Ns. Ylä-servletina toimii LahjalistaServlet.java, johon on sijoitettu yleiskäytössä olevat metodit, kuten listojen haku modelseilta ja JSP-sivun näyttäminen. Muut servlet-luokat perivät tämän luokan.

Näkyminä toimivat jsp-sivut, jotka on sijoitettu kansioon web/. Sovellus hyödyntää bootstrap-kirjastoa, johon liittyvät javascript-koodi on sijoitettu tiedostoon web/js/bootstrap.js. Itse toteutetut javascript-koodi löytyy tiedostosta web/js/omat.js.



## 6 Käyttöliittymä ja järjestelmän komponentit



Sovelluksessa on navigaatiopalkit sekä perussivulla, että kirjautumista vaativilla sivuilla. Sovellus myös hyödyntää Bootstrap-kirjaston Modal-popuppeja, mutta nämä on kuviossa merkitty omiksi sivuikseen selvyiden vuoksi. Modal-sivuihin merkityt kontrollereit ovat niitä kontrollereita, jotka suorittavat muokkaus/poisto-operaatiot tietokantaan.

## 7 Asennustiedot

Asenna sovellus kopiomailla tiedostot internetiin näkyvään kansioon.

Aja "sql/create-tables.sql" tietokantaan. Lisää lisäksi ainakin yksi Ylläpitäjä tietokantaan, tai aja tiedosto "sql/add-test-data.sql".

Aseta tietokannan asetukset oikeaksi tiedostoon web/META-INF/context.xml. Repositorioon ei ole sisällytetty ko. tiedostoa, vaan se on korvattu tiedostolla context.xml.dist jossa alkuperäinen salasana on korvattu. Korvaa kentät "username" "password" ja "url" tietokantaasi vastaaviksi.

Sovellus on toteutettu käyttämällä PostgreSQL tietokantaa, ja en takaa sovelluksen toimimista vakiona muilla tietokantajärjestelmillä.

## 8 Sovelluksen käyttöohje

### 8.1 Käynnistäminen

Sovelluksen voi käynnistää www-selaimella sivustolta:

<http://t-tomikasu.users.cs.helsinki.fi/Lahjalista/lahjalista> .

Jos sovellus on lisätty omalla palvelimella, toimii se osoitteessa

<http://tahan.oman.palvelimen.osoite/Lahjalista/lahjalista>

Ylläpitäjän käyttäjätunnukset ovat oletuksena käyttäjätunnus: "admin" salasana: "password". Omalla palvelimella ne kannattaa vaihtaa omiksi add-test-data.sql -tiedostosta tai suoraan omasta SQL-tietokannasta.

## **9 Testaus, tunnetut bugit ja puutteet & jatkokehitysideat**

Sovellusta on testattu kohtuullisen paljon, ja testeissä on keskitytty erityisesti tietokantaan ajettavien syötteiden tarkistamiseen. Sovelluksen ei pitäisi hyväksyä virheellisiä syötteitä, jotka johtavat SQL-kyselyn virhetilanteeseen. Syötteiden tarkistuksessa on myös varauduttu SQL-injektio ja javascript-hyökkäyksiin.

Ohjelmassa ei käsittääkseni ole vakavia bugeja, jotka vaikuttaisivat ratkaisevasti ohjelman toimintaan. Tiedossa oleva puute on ääkkösten toimimattomuus lahjoja / varauksia lisättäessä. Ääkköset menevät kyllä läpi tietokantaan, mutta javan preparestatement-lausekkeet sotkevat ääkköset ja sovellus ei näytä niitä UTF-8 -mukaisina skandeina. Tämä ei kuitenkaan vaikuta sovelluksen toimintaan, vaan on lähinnä esteettinen haitta.

Jatkokehityksenä sovellukseen voisi lisätä ominaisuuden, että vieraat voivat lisätä samoja lahjoja monta kertaa / eri määriä kuin yksi. Varausten "määrä" -attribuuttia ei ole toteutettu tällä hetkellä. En kuitenkaan pidä tätä suurena puutteena, ja sen hyödyt tulisivat esiin kohtuu harvoissa tilanteissa.

## **10 Omat kokemukset**

### **10.1 HTML**

Bootstrap-kirjasto oli erinomainen apu sovelluksen tekemisessä ja auttoi tekemään sovelluksesta helposti kohtuullisen ammattimaisen näköisen. Myös Bootstrapin Modalit tekivät sovelluksesta selkeämmän, mutta ne vaikeuttivat huomattavasti sovelluksen toiminnan tekemistä, koska sivun ja modalin välillä ei ole Servlet-kontrolleria. Täten kannattaa vastaavassa sovelluksessa harkita, ovatko Modalit todella tarpeellisia.

### **10.2 Java**

Vaikka Servlettien toimintaan oli alkuun hieman vaikeaa päästä sisään, oli kuitenkin loppujenlopuksi miellyttävää tehdä sovellus Javalla. Monet kanssaopiskelijat, jotka tekivät sovelluksen muilla kielillä tuskailivat MVC-mallin kanssa. Omasta mielestä se tuli kuitenkin lähes itsestään Javan kanssa. Toki Javalla joutui kirjoittamaan ”kohtuullisen” paljon koodia näin pientä sovellusta varten, mutta erittäin harva ongelma kurssin aikana koski Javaa ja sen toimintaa.