|  |
| --- |
| Helsingin Yliopisto |
| Lahjalistasovellus |
| Tietokantasovellus, Syksy 2014 |
|  |
| **Tomi Kasurinen** |
|  |

# **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.

# **Yleiskuva järjestelmästä**

## **Käyttäjäryhmät**

**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.

## **Käyttötapaukset**

### **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

### **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.

### **Käyttötapauskaavio**

# **Järjestelmän tietosisältö**

## **Käsitetason tietokantamalli**

## 

## **Tietokohteiden määrttely**

## **Lahjaehdotus**

# 

## **Vieras**

## 

## **Varaus**

## 

## **Ylläpitäjä**

## 

# **Relaatiotietokantakaavio**

# **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-servlettina toimii LahjalistaServlet.java, johon on sijotettu 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/bootsrap.js. Itse toteutetut javascript-koodi löytyy tiedostosta web/js/omat.js.

# **Käyttöliittymä ja järjestelmän komponentit**

## 

Kirjautumista vaativat sivut

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

# **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.

# **Sovelluksen käyttöohje**

## **Käynnistäminen**

# Sovelluksen voi käynnistää www-selaimella sivustolta: <http://t-tomikasu.users.cs.helsinki.fi/Lahjalista/lahjalista> .

# Jos sovellus on lisätty omalla palvelimell, toimii se osoitteessa [http://tahan.oman.palvelimen.osoite/Lahjalista/lahjalista](http://tahan.oma.palvelin.fi/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.

## **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” -atribuuttia ei ole toteutettu tällä hetkellä. En kuitenkaan pidä tätä suurena puutteena, ja sen hyödyt tulisivat esiin kohtuu harvoissa tilanteissa.

# **Omat kokemukset**

## **HTML**

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

## **Java**

Vaikka Servlettien toimintaan oli alkuun hieman vaikeaa päästä siisään, oli kuitenkin loppujenlopuksi miellyttävää tehdä sovellus Javalla. Monet kanssaopiskelijat, jotka tekivät sovelluksun 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.