

HELSINGIN YLIOPISTO

TAPAHTUMATIE TOKANTA

Tietokantasovellus –kurssi, syksy 2012

Topias Rusanen

12/21/2012

Johdanto	3
Toteutus- ja toimintaympäristö	3
Käyttötapauskaavio	4
Käyttötapaukset	4
Käyttäjätilin luominen	4
Kirjautuminen	4
Tapahtumien selaaminen	5
Tapahtumaan osallistuminen ja osallistumisen peruminen	5
Tapahtumien kommentointi	5
Tapahtumien luominen	5
Omien tapahtumien poisto	5
Tietosisältö	5
Relaatiotietokantakaavio	7
Järjestelmän yleisrakenne	8
Järjestelmän komponentit	8
Mallit (Models)	9
Kyselyt (Queries)	9
Servletit (Servlets)	9
Käyttöliittymä	10
Asennustiedot	11
Käynnistys- ja käyttöohje	11

Johdanto

Tietokantasovellus-kurssille tein nettipalvelun, johon kirjautunut käyttäjä pystyy luomaan tapahtumia, kommentoimaan niitä ja liittymään muiden luomiin tapahtumiin. Palvelu on tarkoitettu lähinnä oman jalkapallojoukkueeni sekä muunkin kaveriporukan käyttöön. Yleensä kaveriporukan tapahtumat sovitaan Facebookissa, mutta muutama ihminen ei halua liittyä palveluun, sillä he vastustavat Facebookin tapaa käsitellä käyttäjiltä keräämiään tietoja. Näille henkilöille joudutaan aina erikseen ilmoittamaan esimerkiksi puhelimella, joten tarkoituksena oli luoda palvelu, jonka käyttäjillä ei ole pelkoa tietojensa väärinkäytöstä. Nähtäväksi jää, kuinka paljon tapahtumatietokantaa tullaan käyttämään.

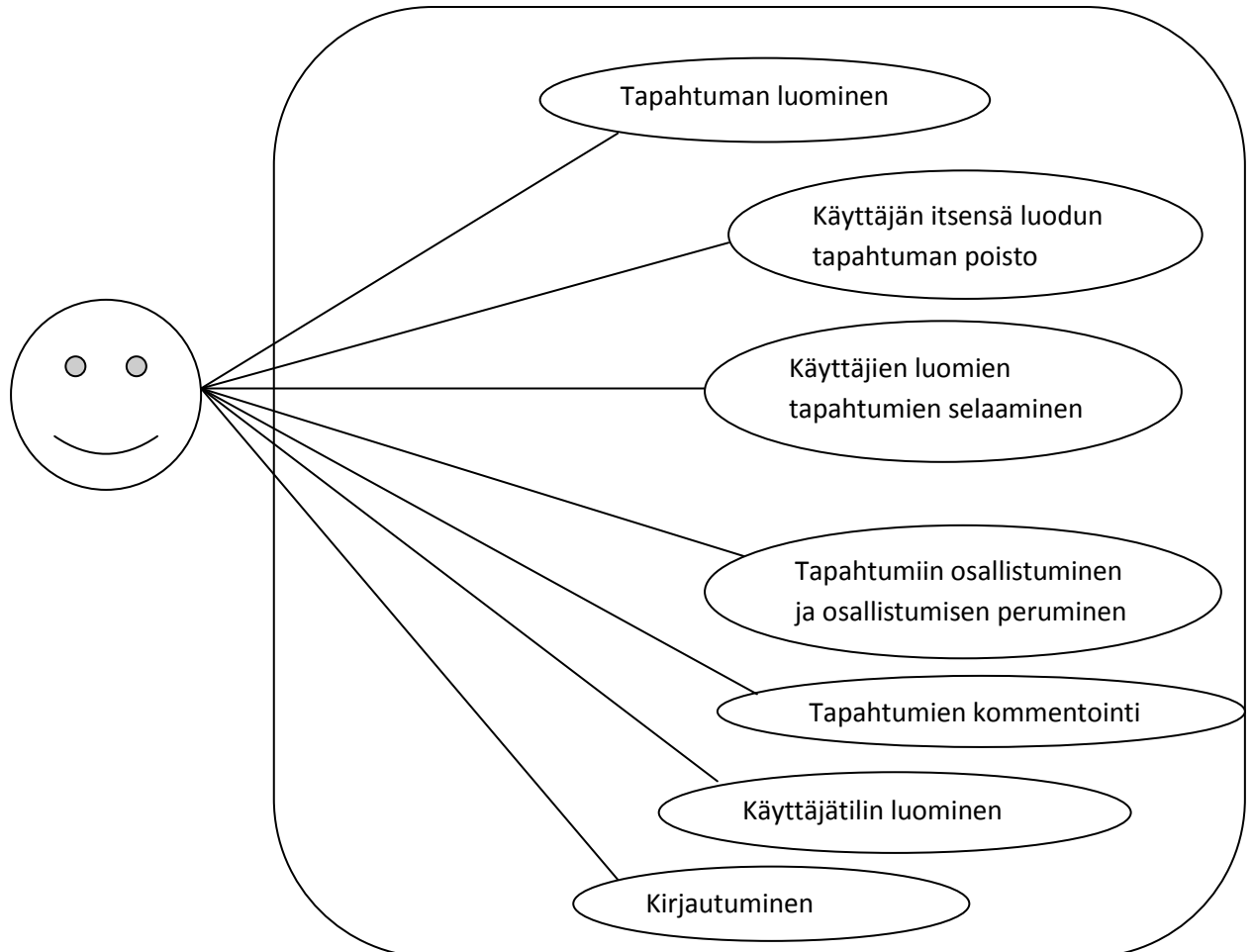
Toteutus- ja toimintaympäristö

Työ on toteutettu käyttämällä Java-servlettejä Tomcat-palvelimessa. Tietokannaksi valittiin PostgreSQL. Kehitysympäristönä toimi NetBeans 7.2 sekä GitHub. Koodin sekä muunkin kehityksen kommentointi tapahtui GitHubin Pull Requesteja käyttämällä. Aina uuden toiminnon toteuttamiseksi luotiin uusi haara, joka toiminnon valmistuttua yhdistettiin master-haaraan.

Toimintaympäristöksi riittää siis mikä tahansa javaa tukeva palvelinalusta. Esimerkiksi työn testaamiseen käytettiin NetBeansin omaa GlassFish-palvelinta, jolla se myös toimi mainiosti. Servlettien luomat sivut ovat puhdasta HTML-tyyppistä kieltä, joten palvelun pitäisi toimia kaikissa selaimissa. Ainakin Firefox, Opera ja Chrome näyttivät sivut moitteetta. Palvelua ei kokeiltu eri tietokannoilla, mutta periaatteessa sen pitäisi melko pienin muutoksin myös onnistua, joskin tämä vaatii muutoksia lähdekoodiin. Tietokanta-ajuri pitää vaihtaa sekä uuden ajurin kirjasto lisätä esimerkiksi Maveniin. Ajurin vaihtamiseen tarvitsee muuttaa vain yhtä koodiriviä.

Käyttötapauskaavio

Käyttötapauskaavio on melko yksinkertainen, sillä minkäänlaisia erityispalveluja tapahtumatietokanta ei tarjoa eri käyttäjille. Käytännössä siis palvelulla on vain yksi yleinen käyttäjä.



Käyttötapaukset

Käyttäjätilin luominen

Käyttäjä luo itselleen tilin antamalla käyttäjänimen ja salasanan. Käyttäjänimi on uniikki tunniste, joten jos käyttäjänimi on käytössä, tapahtuu virhe.

Kirjautuminen

Käyttäjä syöttää kirjautumissivulla käyttäjänimensä ja salasansansa ja painaa tämän jälkeen ”Log in” – nappia. Jos käyttäjänimi ja salasan täsmäivät, käyttäjä ohjataan omalle sivulleen. Muissa tapauksissa näytetään virheviesti.

Tapahtumien selaaminen

Käyttäjälle näytetään kaikki luodut tapahtumat. Tapahtuman linkkiä klikkaamalla käyttäjä ohjataan tapahtuman sivulle.

Tapahtumaan osallistuminen ja osallistumisen peruminen

Tapahtuman sivulla käyttäjä voi painaa "Attend event"-nappia, jolloin hänet kirjataan osallistuvaksi kyseiseen tapahtumaan ja hänen nimensä näkyy tapahtumaan osallistuvien listassa. Jos käyttäjä on jo ilmoittautunut tapahtumaan, "Attend event"-napin tilalla on "Unattend event"-nappi, jota painamalla käyttäjä peruu osallistumisensa.

Tapahtumien kommentointi

Tapahtuman sivulla on kommenttilaatikko, johon käyttäjä voi kirjoittaa kommentin. Painamalla "Comment"-nappulaa kommentti tallentuu tietokantaan ja näytetään tapahtuman sivulla.

Tapahtumien luominen

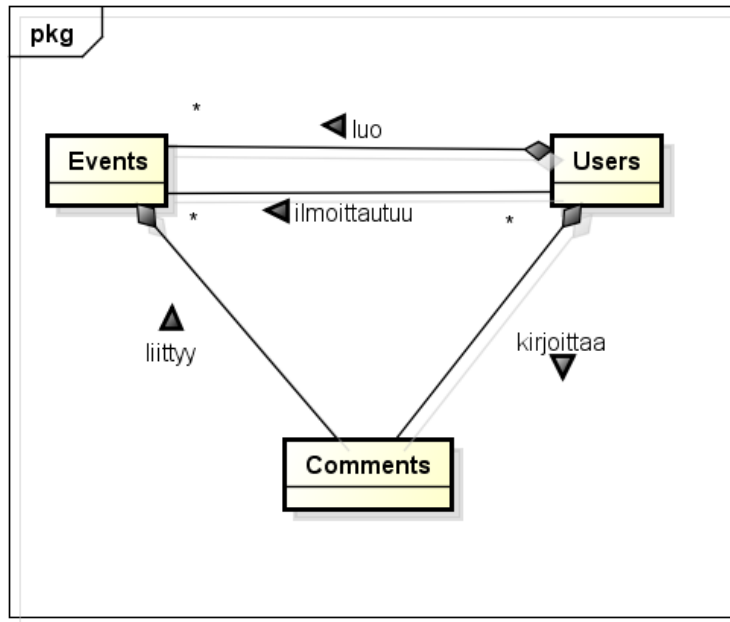
Käyttäjän omalla sivulla on linkki lomakkeeseen, jolla voi luoda oman tapahtuman. Käyttäjä syöttää nimen tapahtumalleen ja antaa mahdollisesti jotain muuta lisätietoa, minkä jälkeen "Create event"-nappia painamalla tapahtuma luodaan tietokantaan ja käyttäjä ohjataan omalle sivulleen.

Omien tapahtumien poisto

Käyttäjän itsensä luomien tapahtumien sivuilla löytyy linkki "Cancel this event". Tätä painamalla käyttäjä poistaa luomansa tapahtuman ja hänet ohjataan omalle sivulleen.

Tietosisältö

Tapahtumatietokanta sisältää kolme erilaista tietotyyppiä: käyttäjän, tapahtuman ja kommentin. Yhteyksiä käyttäjä-luo-tapahtuman, käyttäjä-kirjoittaa-kommentin ja kommentti-liittyy-tapahtumaan ei ole erikseen määritelty, sillä ne ovat monen-suhde-yhteen yhteyksinä melko selkeitä kaavion perusteella ja niillä ei ole omia attribuutteja. Tarkemmat määrittelyt muille löytyvät alta.



powered by Astah

Käyttäjä (User)		
Attribuutti	Arvojoukko	Kuvailu
userName	Merkkijono	Käyttäjän käyttäjänimi
password	Merkkijono	Käyttäjän salasana
name	Merkkijono	Käyttäjän oikea nimi
joinDate	Päivämäärä	Päivämäärä, jolloin käyttäjä on liittynyt palveluun

Tapahtuma (Event)		
Attribuutti	Arvojoukko	Kuvailu
createdDate	Päivämäärä	Aika, jolloin tapahtuma on luotu
name	Merkkijono	Tapahtuman nimi
information	Merkkijono	Lisätietoja tapahtumasta

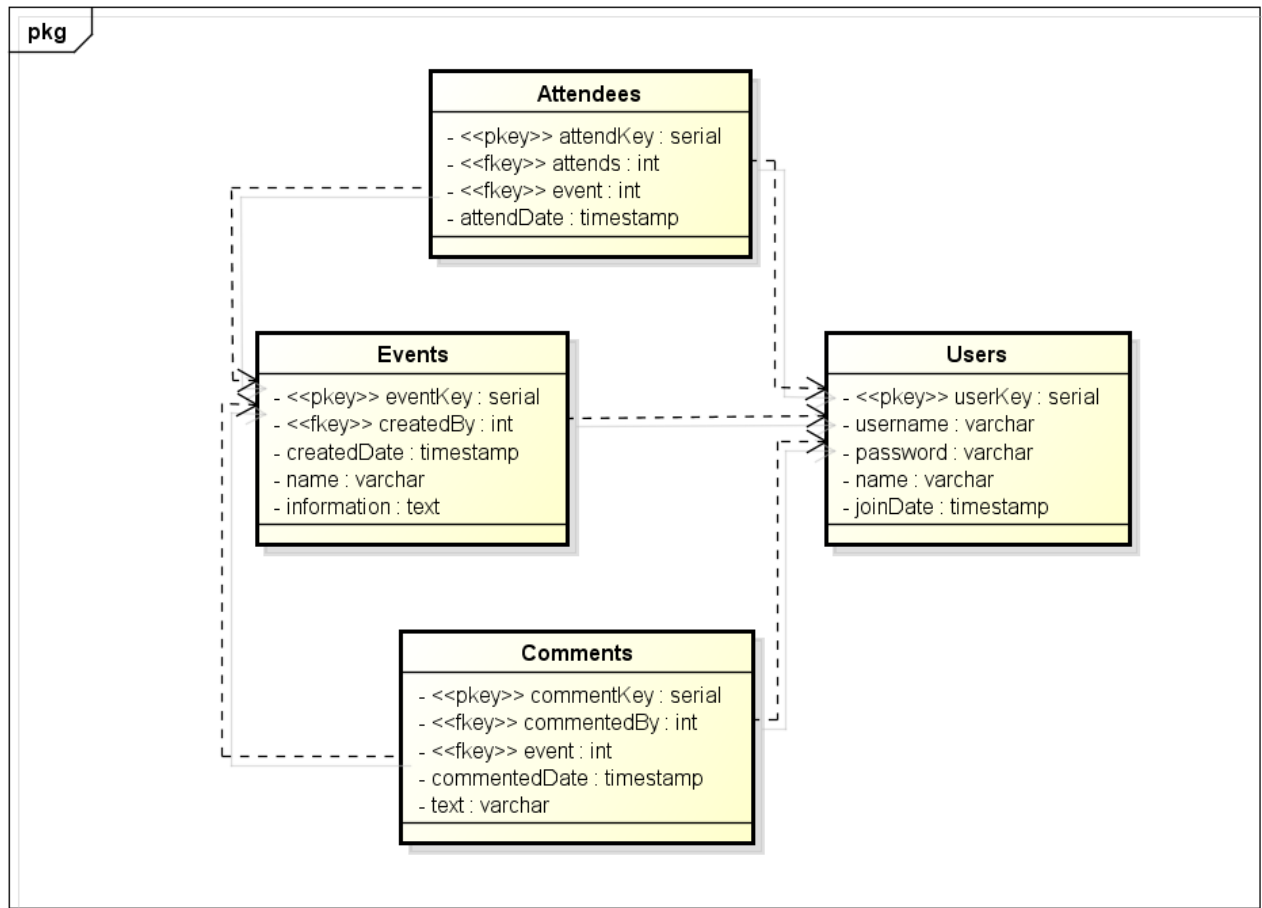
Kommentti (Comment)		
Attribuutti	Arvojoukko	Kuvailu
commentedDate	Päivämäärä	Aika, jolloin kommentti on lisätty
text	Merkkijono	Kommentin teksti

Kommentti (Comment)		
Attribuutti	Arvojoukko	Kuvailu
commentedDate	Päivämäärä	Aika, jolloin kommentti on lisätty
text	Merkkijono	Kommentin teksti

Käyttäjä-ilmoittautuu-tapahtumaan -yhteys		
Attribuutti	Arvojoukko	Kuvailu
attendDate	Päivämäärä	Aika, jolloin ilmoittautuminen on tapahtunut

Relaatiotietokantakaavio

Seuraavassa kaaviossa esitellään tietokannan taulut, niiden pää- ja viiteavaimet sekä muut attribuutit ja taulujen väliset yhteydet. Kaavio on tehty tietosisältökaavion pohjalta ja siihen on lisätty taulujen surrogaattiavaimet sekä viiteavaimet. SQL-muotoiset Create table –lauseet löytyvät kaavion jälkeen.



powered by Astah

```

CREATE TABLE Users
(
  userKey serial NOT NULL PRIMARY KEY,
  userName varchar NOT NULL UNIQUE,
  password varchar NOT NULL,
  name varchar NOT NULL,
  joinDate timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
);
  
```

```

CREATE TABLE Events
(
  eventKey serial NOT NULL PRIMARY KEY,
  createdBy int references Users(userKey),
  createDate timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  name varchar NOT NULL,
  information text
);
  
```

```

CREATE TABLE Attendees
(
  attendKey serial NOT NULL PRIMARY KEY,
  attends int references Users(userKey),
  
```

```

event int references Events(eventKey),
joinDate timestamp DEFAULT CURRENT_TIMESTAMP,
UNIQUE (attends, event)
);

CREATE TABLE Comments
(
commentKey serial NOT NULL PRIMARY KEY,
commentedBy int references Users(userKey),
event int references Events(eventKey),
commentedDate timestamp DEFAULT CURRENT_TIMESTAMP,
text varchar NOT NULL
);

```

Järjestelmän yleisrakenne

Järjestelmän rakentamisessa on käytetty MVC-mallia pohjana. Mallit löytyvät nimensä mukaisesti nimettynä Models-kansiosta. Kontrollerit eli tietokantakyselyt löytyvät Queries-hakemistosta ja näkymät eli servletit löytyvät Servlets-hakemistosta. Käytetyt kirjastot löytyvät hakemistosta WEB-INF/lib/. Kaikki tiedostonimet alkavat pienellä alkukirjaimella paitsi Java-luokat.

Järjestelmän komponentit

bottom.jspf: Jokaisen sivun loppuun lisättävä tiedosto, joka sisältää ohjelman nimen, tekijän nimen ja kurssin, jolle ohjelma on tehty

createEvent.jsp: Sisältää tapahtuman luomiseen tarvittavan lomakkeen.

createUserPage.jsp: Sisältää uuden käyttäjätilin luomiseen tarvittavan lomakkeen.

eventPage.jsp: Jokaisen tapahtuman sivun luomiseen käytettävä pohja. Sisältää muunmuassa tapahtuman nimen, tapahtuman poistamiseen käytettävän linkin, tapahtumaan ilmoittautumiseen ja ilmoittautumisen perumiseen käytettävät linkit, ilmoittautuneiden nimet sekä kommentit.

events.jsp: Sivu, josta löytyvät kaikki tapahtumat.

index.jsp: Aloitusivu, joka ohjaa kirjautumissivulle.

loginpage.jsp: Sivu, joka sisältää kirjautumiseen tarvittavan lomakkeen sekä linkin uuden käyttäjätilin luomiseen käytettävään sivuun.

top.jspf: Ylävalikko, joka lisätään kirjautumista vaativien sivujen alkuun. Sisältää linkit kaikkien tapahtumien listaan ja käyttäjän omaan sivuun sekä uloskirjautumiseen käytettävän linkin. Kertoo myös merkistökoodauksen olevan UTF-8.

userPage.jsp: Käyttäjän oma sivu, joka tervehtii käyttäjää ja näyttää käyttäjätunnuksen, käyttäjän luodut tapahtumat, tapahtumat joihin käyttäjä osallistuu sekä linkin uuden tapahtuman luomiseen.

WEB-INF/web.xml: Tässä tiedostossa määritellään, mitkä osoitteet kutsuvat mitäkin servlettiä. Sisältää myöskin yleisiä palvelimen määrittelyjä.

WEB-INF/lib/jstl-1.2.jar: JavaServer Pages Standart Tag Library, tämä kirjasto tuo tuen JavaServer Pages-sivuille ja niiden tarjoamalle toiminnallisuudelle.

WEB-INF/lib/postgresql-9.1-901-1.jdbc4.jar: PostgreSQL-tietokannan ajuri.

Tässä vielä servlettien määrittelyt:

Mallit (Models)

Comment: Kommentin malli, joka sisältää kommentin tunnuksen, luontiajan, kommentoijan tunnuksen sekä itse tekstin.

DatabaseConnection: Tietokantaluokka, joka sisältää itse tietokantayhteyden ja ajurin määrittelyt. Kaikki kyselyluokat ovat tämän luokan aliluokkia.

Event: Tapahtuman malli, jota käytetään tapahtumiin liittyvien tietojen tallentamiseen sekä niiden muokkaamiseen ja hakemiseen tietokannasta. Sisältää ainakin tapahtuman tunnuksen, nimen sekä tapahtuman luojaan tunnuksen. Lisäksi sisältää metodit tapahtuman luojaan hakemiseen, tapahtuman tietojen hakemiseen, osallistujien hakemiseen sekä tapahtumaan liittyvien kommenttien hakemiseen.

User: Käyttäjän malli, jota käytetään käyttäjään liittyvien tietojen tallentamiseen sekä niiden muokkaamiseen ja hakemiseen tietokannasta. Sisältää käyttäjän tunnuksen ja nimen. Lisäksi sisältää metodit käyttäjän luomien tapahtumien hakemiseen, käyttäjän osallistumien tapahtumien hakemiseen, tapahtumaan ilmoittautumiseen ilmoittautumisen perumiseen ja käyttäjän sisäänkirjaamiseen.

Kyselyt (Queries)

CommentQuery: Sisältää kommentteihin liittyvien tietokantakyselyiden logiikan. Tällä hetkellä luokalla on vain yksi metodi, kommenttin luominen.

EventQuery: Sisältää tapahtumiin liittyvien tietokantakyselyiden logiikan. Sisältää metodit tapahtuman hakemiseen tietokannasta, kaikkien tapahtumien hakemiseen, tapahtuman tietojen hakemiseen, tapahtuman kommenttien hakemiseen, tapahtumaan ilmoittautuneiden käyttäjien hakemiseen, tietyn käyttäjän luomien tapahtumien hakemiseen, tietyn käyttäjän ilmoittautuneiden tapahtumien hakemiseen sekä tapahtuman luomiseen, poistamiseen ja tapahtuman osallistujien ja kommenttien poistamisen käytettävät metodit.

UserQuery: Sisältää käyttäjiin liittyvien tietokantakyselyiden logiikan. Sisältää metodi käyttäjän hakemiseen tunnuksen ja nimen perusteella sekä käyttäjän luomiseen, tapahtumaan ilmoittautumiseen sekä ilmoittautumisen perumiseen käytetyt metodit.

Servletit (Servlets)

AttendEventServlet: Tämä servlet ilmoittaa käyttäjän tapahtumaan ja ohjaa sen jälkeen tapahtuman sivulle.

CreateEventServlet: Tämä servlet näyttää tapahtuman luomiseen käytetyn lomakkeen ja käsittelee täytetyn lomakkeen luoden uuden tapahtuman tietokantaan.

CreateUserServlet: Tämä servlet näyttää uuden käyttäjätilin luomiseen käytettävän lomakkeen ja käsittelee täytetyn lomakkeen luoden uuden käyttäjätilin. Tilin luonnin jälkeen käyttäjä kirjataan sisään kyseisellä tilillä ja ohjataan omalle sivulleen.

DeleteEventServlet: Tämä servlet poistaa attribuuttina annetun tapahtuman, jos sisäänkirjautunut käyttäjä on tapahtuman luoja. Tämän jälkeen se ohjaa käyttäjän takaisin omalle sivulleen.

EventListServlet: Tämä servlet näyttää kaikki tapahtumat listana ja luo linkit niiden sivuille.

EventPageServlet: Tämä servlet näyttää tapahtuman sivun ja sisältää lomakkeen tapahtuman kommentoimiseen. Lisäksi se käsittelee kommenttilomakkeen ja lisää kommentin tietokantaan.

LoginServlet: Tämä servlet näyttää kirjautumislomakkeen ja käsittelee sen.

LogoutServlet: Tämä servlet kirjaa käyttäjän ulos poistamalla sessio-olion palvelimelta.

MainServlet: Tämä on kaikkien servlettien yliluokka, joka sisältää ainoastaan metodin, joka tarkistaa, että käyttäjä todella on kirjautunut sisään.

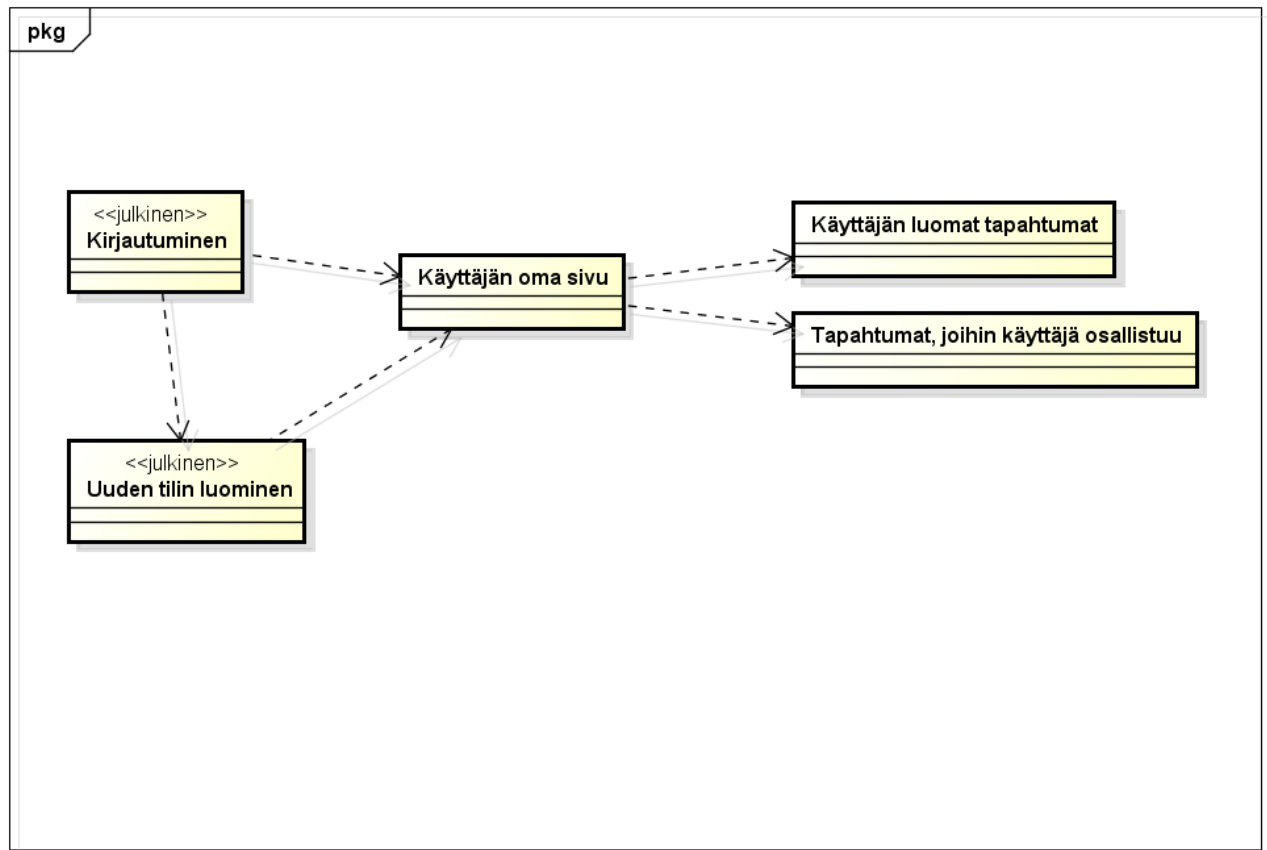
UnattendEventServlet: Tämä servlet sisältää peruu käyttäjän ilmoittautumisen tapahtumaan ja ohjaa sen jälkeen käyttäjän kyseisen tapahtuman sivulle.

UserPageServlet: Tämä servlet tarkistaa vain, onko käyttäjä kirjautunut sisään ja sen jälkeen ohjaa käyttäjän sen omalle sivulle.

Käyttöliittymä

Käyttöliittymä on toteutettu linkeillä sivuilta toiselle sekä sivuston yläreunassa löytyvällä navigointipalkilla, josta on linkit listaan kaikista tapahtumista, käyttäjän omalle sivulle sekä uloskirjautumiseen.

Navigointipalkki löytyy ainoastaan sivuilta, jotka vaativat kirjautumisen. Alla kuvaus, miltä sivuilta löytyy linkki millekin sivulle, navigointipalkkia ei oteta huomioon.



powered by Astah

Asennustiedot

Sovellus on tällä hetkellä asennettu laitoksen users-palvelimelle ja löytyy osoitteesta `t-trusanen.users.cs.helsinki.fi`. Users-palvelimelle on tilattu Tomcat-palvelinympäristö ja PostgreSQL-tietokanta.

Uuteen ympäristöön asennettaessa ei pitäisi tehdä muuta kuin siirtää `.war` arkisto uuteen javaa tukevaan palvelinympäristöön ja muuttaa lähdekoodista `DatabaseConnection`-luokasta tietokannan osoite. Jos tietokanta on jokin muu kuin PostgreSQL, täytyy vielä vaihtaa lähdekoodista `DatabaseConnection`-luokasta tietokanta-ajuri sekä asentaa uusi kirjasto.

Käynnistys- ja käyttöohje

Tapahtumatietokanta löytyy osoitteesta `t-trusanen.users.cs.helsinki.fi`. Testitunnuksina voi käyttää esimerkiksi tunnusta "Topias" tai "Erkki". Molempien salasana on "uliuli". Vaihtoehtoisesti voi myös luoda oman tunnuksen, eri käyttäjillä ei ole erilaisia toimintoja.