

Tietokantasovellusdokumentaatio

Janne Ruoho

Sisällysluettelo

1. Johdanto.....	2
1.1 Järjestelmän idea ja rakenne.....	2
1.2 Toteutusympäristö.....	2
1.3 Toimintaympäristö.....	2
2. Yleiskuva järjestelmästä.....	2
2.1 Käyttötapauskaavio.....	2
2.2 Käyttäjryhmät.....	3
2.3 Käyttötapauskuvaukset.....	3
3. Järjestelmän tietosisältö.....	4
Käsitekaavion tietokohteet.....	4
4. Relaatiotietokantakaavio.....	6
5. Järjestelmän yleisrakenne.....	6
6. Käyttöliittymä ja järjestelmän komponentit.....	7
7. Asennustiedot.....	7
8. Käyttöohje.....	8
9. Testaus, tunnetut bugit ja puutteet.....	8
10. Omat kokemukset.....	8
11. Liitteet.....	8

1. Johdanto

1.1 Järjestelmän idea ja rakenne

Työn aiheena on luoda **monikäyttöinen keskustelufoorumi**. Keskustelufoorumin rekisteröitynyt jäsen voi kirjaututtuaan lukea ja luoda uusia viestiketjuja, vastata muiden viesteihin viestiketjussa, sekä muokata omia viestejään. Jäsen voi myös hakea viestejä foorumin hakutoiminnolla kirjoittajan nimen, viestiketjun otsikon, viestin sisällön tai iän perusteella. Rekisteröimättömät vieraat voivat vain selata julkisia viestiketjuja ja lukea niiden viestejä.

Foorumin ylläpitäjät pystyvät lisäämään, poistamaan ja muokkaamaan jäseniä sekä hallinnoimaan kaikkia viestiketjuja ja viestejä. Ylläpitäjät pystyvät myös hallinnoimaan jäsenryhmiä, joille voidaan esimerkiksi määrittää ryhmän sisäisiä viestiketjuja. Jokainen jäsen voi kuulua useaan jäsenryhmään.

1.2 Toteutusympäristö

Sovellus toteutetaan käyttäen ohjelmointikielenä PHP:ta ja tietokantana PostgreSQL:ää.

1.3 Toimintaympäristö

Sovellus toimii laitoksen users-palvelimella. Sen osoite on <http://jmruoho.users.cs.helsinki.fi/tsoha>.

Sovellus tarvitsee toimiakseen Apache-palvelimen vähintään PHP-versiolla 5.3 sekä PostgreSQL-tietokannan. Käyttäjän selaimen tulee tukea HTML5-standardia sekä Javascriptiä.

2. Yleiskuva järjestelmästä

2.1 Käyttötapauskaavio

Alla olevassa käyttötapauskaaviossa on lueteltu kaikki käyttötapaukset. Kaikki käyttötapaukset, mitkä ovat jäsenen käytettävissä, ovat myös ylläpitäjän käytettävissä; viivat on jätetty merkitsemättä tilan säästämiseksi.



2.2 Käyttäjärühmät

Vierailija

Vierailijalla tarkoitetaan ketä tahansa ei-rekisteröitynyttä käyttäjää. Kaikki muut sidosryhmät kuuluvat tähän sidosryhmään.

Jäsen

Jäsenellä tarkoitetaan ketä tahansa keskustelufoorumille rekisteröityä käyttäjää. Myös kaikki ylläpitäjät kuuluvat tähän sidosryhmään.

Ylläpitäjä

Ylläpitäjällä tarkoitetaan foorumin jäsentä, jolla on kaikki oikeuden foorumin hallintaan.

2.3 Käyttötapauskuvaukset

Vierailijan käyttötapauskuvaukset

- Viestiketjujen selaaminen: Kuka tahansa voi selata foorumin viestiketjuja, mikäli ne on määritetty julkisiksi.
- Viestien lukeminen: Kuka tahansa voi lukea julkisen viestiketjun viestejä.
- Rekisteröityminen: Kuka tahansa voi rekisteröityä foorumille, ellei ole jo rekisteröitynyt

Jäsenen käyttötapauskuvaukset

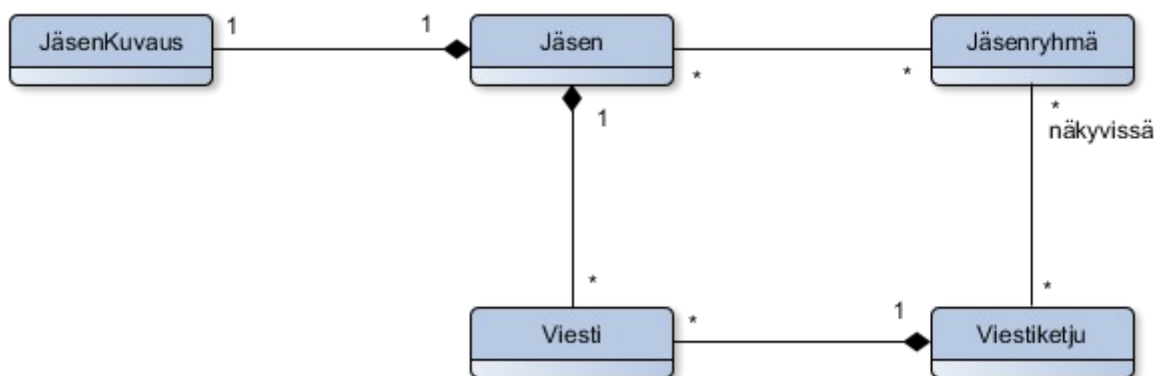
- Omien tietojen muokkaaminen: Muokattavia tietoja ovat mm. sähköpostiosoite, salasana.
- Hakutoiminnon käyttäminen: Hakutoiminnolla voi hakea viestejä viestiketjun otsikon, viestin sisällön, kirjoittajan nimen ja viestin iän perusteella.
- Viestin kirjoittaminen ja muokkaaminen: Jäsen voi kirjoittaa viestin mihin tahansa viestiketjuun, jos jäsen kuuluu jäsenryhmään jolle on annettu oikeudet kyseiseen viestiketjuun. Viesti voi olla myös vastaus mihin tahansa muuhun viestiketjun viestiin. Jäsen pystyy myös muokkaamaan omia viestejään.
- Viestiketjun aloittaminen: Jäsen voi aloittaa kokonaan uuden viestiketjun.
- Muita käyttötapauksia: kirjautuminen

Ylläpitäjän käyttötapauskuvaukset

- Jäsenryhmien muokkaaminen: Jäsenryhmien lisääminen ja poistaminen, sekä jäsenten lisääminen ja poistaminen jäsenryhmiin.
- Jäsenlistan muokkaaminen: Jäsenten rekisteröiminen ja poistaminen
- Jäsenen tietojen muokkaaminen: Sama kuin jäsenen käyttötapauskuvauksessa 'omien tietojen muokkaaminen', mutta ylläpitäjä voi muokata kenen tahansa tietoja

- Minkä tahansa viestin muokkaaminen/poistaminen: Ylläpitäjä voi muokata ja poistaa kenen tahansa viestejä.
- Viestiketjun näkyvyyden asettaminen: Ylläpitäjä voi tarvittaessa asettaa viestiketjun näkyväksi vain tietylle jäsenryhmälle.

3. Järjestelmän tietosisältö



Kuvassa on kuvattu järjestelmän tietosisältö käsitetasen kaaviolla.

Käsitekaavion tietokohteet

Tietokohde: Jäsen

Attribuutti	Arvojoukko	Kuvailu
Tunnus	Merkkijono	Kirjautumistunnus
Salasana	Merkkijono	Kirjautumissalasana (hash)
Suola	Merkkijono	Salasanan generoitu suola (salt)
Ylläpitäjä	Kokonaisluku	1 tai 0 sen mukaan, onko jäsen ylläpitäjä
Suljettu	Kokonaisluku	1 tai 0 sen mukaan, onko tämä tunnus suljettu (porttikielto)

Jäsenellä on tunnus ja salasana, sekä se joko on ylläpitäjä tai ei. Ylläpitäjät siis eivät ole oma jäsenryhmänsä, vaan ylläpitäjäys on jäsenen ominaisuus.

Tietokohde: Jäsenkuvaus

Attribuutti	Arvojoukko	Kuvailu
Rekisteröitymisaika	Kokonaisluku	Käyttäjän rekisteröitymisaika (unix timestamp)
Sähköposti	Merkkijono	Käyttäjän sähköposti

Oikea nimi	Merkkijono	Käyttäjän oikea nimi
Ikä	Kokonaisluku	Käyttäjän ikä
Sukupuoli	Kokonaisluku	1 = mies, 0 = nainen

Jokainen jäsenkuvaus liittyy aina johonkin mutta vain yhteen jäseneen. Jäsenkuvaus sisältää tietoja jäsenestä.

Tietokohde: Jäsenryhmä

Attribuutti	Arvojoukko	Kuvailu
Nimi	Merkkijono	Jäsenryhmän nimi

Jäsenryhmällä on jokin nimi ja siihen voi kuulua määrittämätön määrä jäseniä. Jäsenet voivat myös kuulua useisiin ryhmiin.

Tietokohde: Viesti

Attribuutti	Arvojoukko	Kuvailu
Viestin nro	Kokonaisluku	Viestin numero ketjussa alkaen 1:stä
Vastauksen nro	Kokonaisluku	Sen viestin numero, johon tämä on kirjoitettu vastauksena
Sisältö	Merkkijono	Viestin sisältö
Lähetysaika	Kokonaisluku	Viestin lähetysaika (unix timestamp)

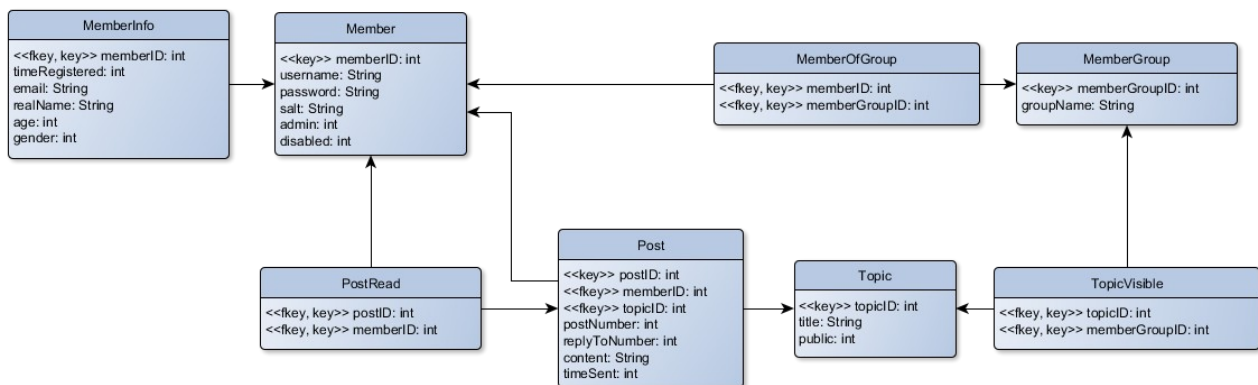
Viestillä on lähetysaika ja sisältö. Viestillä on aina yksi lähettäjä (jäsen) ja viesti kuuluu yhteen viestiketjuun. Viestillä voi myös olla viittaus toiseen viestiin (vastaus).

Tietokohde: Viestiketju

Attribuutti	Arvojoukko	Kuvailu
Otsikko	Merkkijono	Viestiketjun otsikko
Julkinen	Kokonaisluku	1 tai 0 sen mukaan, onko viestiketju julkinen (näkyvissä ei-jäsenille)

Viestiketjulla on aina otsikko. Viestiketju sisältää useita viestejä, ja se voidaan määrittää näkyväksi vain tietyille jäsenryhmille.

4. Relaatiotietokantakaavio



Kuvassa on järjestelmän relaatiotietokantakaavio

5. Järjestelmän yleisrakenne

Sovellusta tehdessä on noudatettu MVC-arkkitehtuuria, joten kaikki luokat on jaoteltu niiden tehtävien mukaisesti kansioihin application/controller, application/model ja application/view tai niiden alikansioihin. Sovelluksen käynnistyskripti sijaitsee tiedostossa index.php ja tarvittavat asetukset defines.php -tiedostossa.

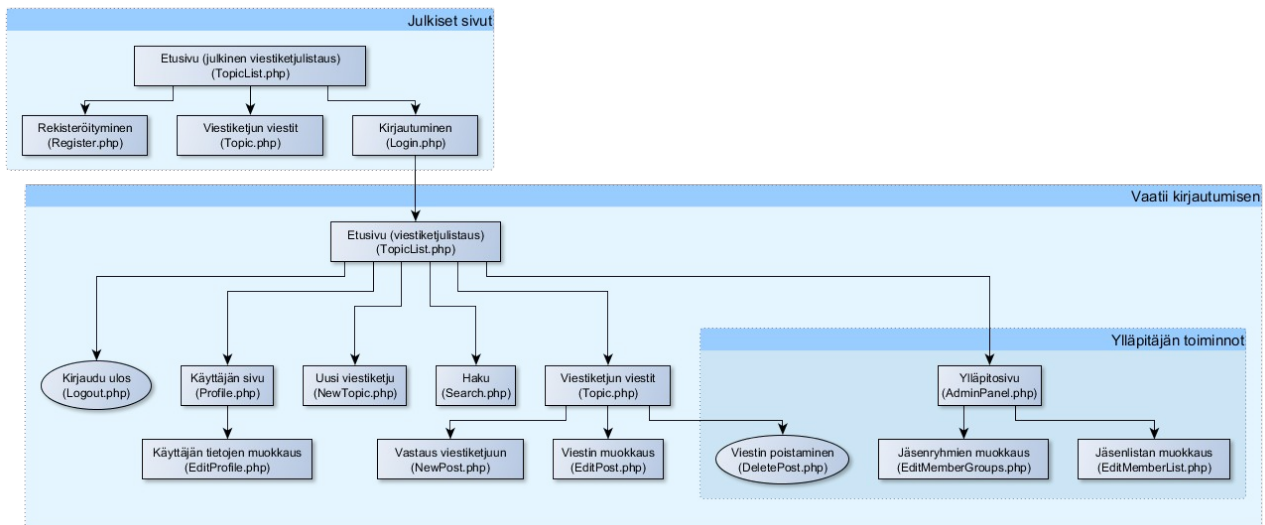
Kaikki käyttäjien mahdolliset toiminnot on sijoitettu kansioon application/controller/action ja ne periytyvät luokasta AbstractAction. Toiminnot otetaan käyttöön lataamalla ne App-luokan addAction-metodissa kutsumalla ActionHandlerin addAction-metodia. AddActionin ensimmäinen parametri on toiminnon nimi, joka määrittää sen URL-osoitteen. Toinen parametri on olio toimintoluokasta. Esimerkiksi toiminnon AdminBanUser, jonka nimeksi antaisimme 'adminbanuser', olisi käytettävissä osoitteesta <http://foorum.in.osoite.com/?action=adminbauser>.

Jotta luokkien automaattinen lataaminen toimisi, jokaisen luokan nimiavaruus on määritetty vastaamaan sen kansiopolkua. Esimerkiksi, jos luodaan uusi toimintoluokka FooBar.php kansioon application/controller/action, niin sen nimiavaruudeksi tulee application\controller\action.

Kaikki luokat nimetään CamelCasella, ja luokan tiedostonimen tulee vastata täysin luokan nimeä. Esim. luokan AdminBanUser tiedosto nimetään AdminBanUser.php:ksi. Näkymät kansiossa application/view on nimetty pienillä kirjaimilla, koska ne eivät sisällä luokkaa eikä niitä siten ladata automaattisesti.

Sovellus käyttää istuntoja (session) käyttäjän pitämiseksi kirjautuneena, sekä jossain tapauksissa säilyttämään käyttäjän syötteen virhetilanteen yhteydessä. Kirjautumisessa istuntoon sisällytetään neljä tietoa: käyttäjänimi, käyttäjän id, ylläpitäjäys ja käyttäjän IP. Listan viimeisin on yksinkertainen (muttei täysin vedenpitävä) suojaus istuntokaappauksia varten.

6. Käyttöliittymä ja järjestelmän komponentit



7. Asennustiedot

Sovellus asennetaan kopioimalla *public*-kansio palvelimen julkiseen kansioon (yleensä *htdocs*, *public_html* tms) tai johonkin sen alikansioon. Kansiot *html-demo* ja *tietokanta* voi poistaa sisältöineen, ne eivät ole tarpeellisia ohjelman käytössä.

Avaa tiedosto *defines.php*, joka löytyy *public*-hakemiston juuresta. Muuta rivi

```
define('BASEURL', 'http://jmruoho.users.cs.helsinki.fi/tsoha/');
```

vastaamaan oman palvelimesi osoitetta, esimerkiksi jompi kumpi seuraavista:

```
define('BASEURL', 'http://minun.omapalvelin.fi/foorumi/');  
define('BASEURL', 'foorumi/'); //myös pelkkä hakemistopolku riittää
```

Osoitteen perässä on oltava */*-merkki. Muuta myös rivi

```
define('DB_DSN', 'pgsql:');
```

vastaamaan tietokantasi DSN:ää, esimerkiksi

```
define('DB_DSN', 'pgsql:dbname=kanta;host=localhost;user=juuseri;password=sala');
```

mikäli tietokantasi vaatii nämä tiedot toimiakseen.

Tietokantataulujen pystyttämiseksi tulee tietokannassa ajaa komennot tiedostosta *create-tables.sql* (ks. liite 1).

Asennuksen jälkeen sovellukseen voi kirjautua tunnuksella **admin** ja salasanalla **admin**. Muista vaihtaa salasana kohdasta *Näytä profiili* > *Muokkaa profiilia*.

8. Käyttöohje

Sovelluksen osoite on <http://jmruoho.users.cs.helsinki.fi/tsoha/>

Pääkäyttäjän tunnus (tunnus, salasana):

admin, admin

Tavallisen käyttäjän tunnus (tunnus, salasana):

Matti Meikäläinen, SalainenSana

9. Testaus, tunnetut bugit ja puutteet

Ei tunnettuja bugeja.

10. Omat kokemukset

Sovelluksen tekeminen oli mielestäni ihan mukavaa, vaikka PHP:n kanssa saa aina tasaisin väliajoin hakata päätään seinään sen tietyistä ominaisuuksista johtuen. Olen aikaisemminkin tehnyt kaikenlaista PHP:lla joten sovellusta tehdessä en oppinut hirveästi uutta, mutta sain kyllä paljon uusia *ahaa-elämyksiä*, ja jos alottaisin sovelluksen alusta, niin tekisin sen varmasti paremmin.

Minulla ei oikeastaan ollut minkäänlaista suunnitelmaa aloittaessani sovellusta, mikä tietenkin kostautui jälkikäteen. Koodista ei tullut yhtään niin kaunista ja selkeää kuin olisin toivonut. Seuraavaan projektini tueksi aion todennäköisesti ottaa jonkin frameworkin, ehkä sillä saisi vähän järjestystä sovelluksen rakenteeseen.

Yleisesti ottaen sovellus kuitenkin toimii ja sitä pystyy muokkaamaan ja laajentamaankin melko helposti, kunhan ensin ymmärtää sovelluksen rakenteen (tai rakenteettomuuden).

11. Liitteet

1. Tietokantataulujen luontiin käytetyt komennot: create-tables.sql
<https://raw.githubusercontent.com/jruo/tsoha/master/sql/create-tables.sql>