

Tietokantasovellus – Dokumentaatio

Rodion "rodde" Efremov

2014, periodi 4

multilog

Sisältö

1	Johdanto	1
2	Yleiskuva järjestelmästä (multilog)	1
2.1	Käyttötapaukset	3
2.2	Käyttäjäryhmät	3
2.3	Käyttötapauskuvaukset	4
3	Käyttöliittymä	5

1 Johdanto

Tämä dokumentti on tarkoitettu kuvaamaan Tietokantasovellus-kurssin (582203) aikana toteutettu verkkosovellus. Ajallisesti ottaen kurssisuoritukseni on ajoitettu kevään 2014 neljännelle periodille. Harjoitustyön aiheena on Keskustelufoorumi ja järjestelmän nimi on **multilog**. multilog ei paljon eroa phpBB:stä, mutta todennäköisesti tulee olemaan yksinkertaisempi ainakin sivujen tyyliytyksen osalta. Kuten yleensäkin, multilog:n on tarkoitus olla helppokäyttöinen ja ilmainen palvelu, jossa käyttäjäkunta pääsee keskustelemaan vapaasti. Toteutuskieleksi olen valinnut Javan (servletit Tomcat:n ajettavaksi), ja tietokannaksi olen valinnut niin ikään suositeltu PostgreSQL. Tarkkaanottaen, toteutus perustuu Java EE 5 spesifikaation, joten ohjelmisto voi ajaa ainakin Tomcat 6:ssa. Saadakseen postausten tyyllittely toimimaan, käyttäjän selaimen on tuettava Javascript. Sovelluksen liittyvät SQL-toiminnot todennäköisesti ei saada standardinmukaisiksi, jotenka järjestelmä on sidottu PostgreSQL:n käyttöön tietokantanaan.

2 Yleiskuva järjestelmästä (multilog)

Aiheenani on keskustelufoorumin toteuttaminen. Järjestelmässä on kolme käyttäjäkategoriaa:

- ylläpitäjä (admin),
- moderaattori (moderator),
- käyttäjä (user).

Käyttäjä voi luoda tunnuksensa rekisteröitymällä sovellukseen. Ensimmäinen ylläpitäjä on luotu SQL-komennosta. Myöhemmin kun järjestelmässä alkaa olla käyttäjiä, ylläpitäjä voi ylentää tavallisen käyttäjän moderaattoriksi tai suoraan ylläpitäjäksi. Ylläpitäjä voi pudottaa moderaattorin peruskäyttäjäksi ja poistaa mielialvaltaisen käyttäjän, jolloin myös moderaattoreiden poisto on mahdollista. Lisäksi, mikä tahansa rekisteröitynyt henkilö voi poistaa profiilinsa. Mitä tulee tilanteseen, jossa ylläpitäjä poistaa itseään, onnistuu se vain jos poiston jälkeen järjestelmään jää vähintään yksi muu ylläpitäjä. (Järjestelmässä pitää olla vähintään yksi ylläpitäjä.) Moderaattori voi poistaa käyttäjien viestit, mikäli siihen on tarvetta. Lisäksi, moderaattori voi asettaa käyttäjät maksimissaan 7 vuorokauden käyttökieltoon. Toisaalta, moderaattori voi ylentää käyttäjän moderaattoriksi. Mitä tulee keskustelusäikeisiin, uusien luonti onnistuu jo peruskäyttäjän oikeuksilla; säikeiden poistaaminen vaatii vähintään moderaattorin oikeudet. Ylläpitäjät pystyvät tekemään samat toiminnot kuin moderaattoritkin.

Mitä tulee itse foorumin rakenteeseen, alkusivulla voi selata "teemoja". Jokainen teema pitää sisällään vähintään yhden "säikeen", joista jälkimmäinen pitää sisällään siihen liittyvän keskusteluhistorian. Teemojen luonti ja poistaminen vaatii ylläpitäjän. Keskustelun lisäksi jokainen käyttäjä/moderaattori/ylläpitäjä voi lähettää yksityiset viestit mielivaltaiselle käyttäjälle ilman statusrajoja. Sovellus

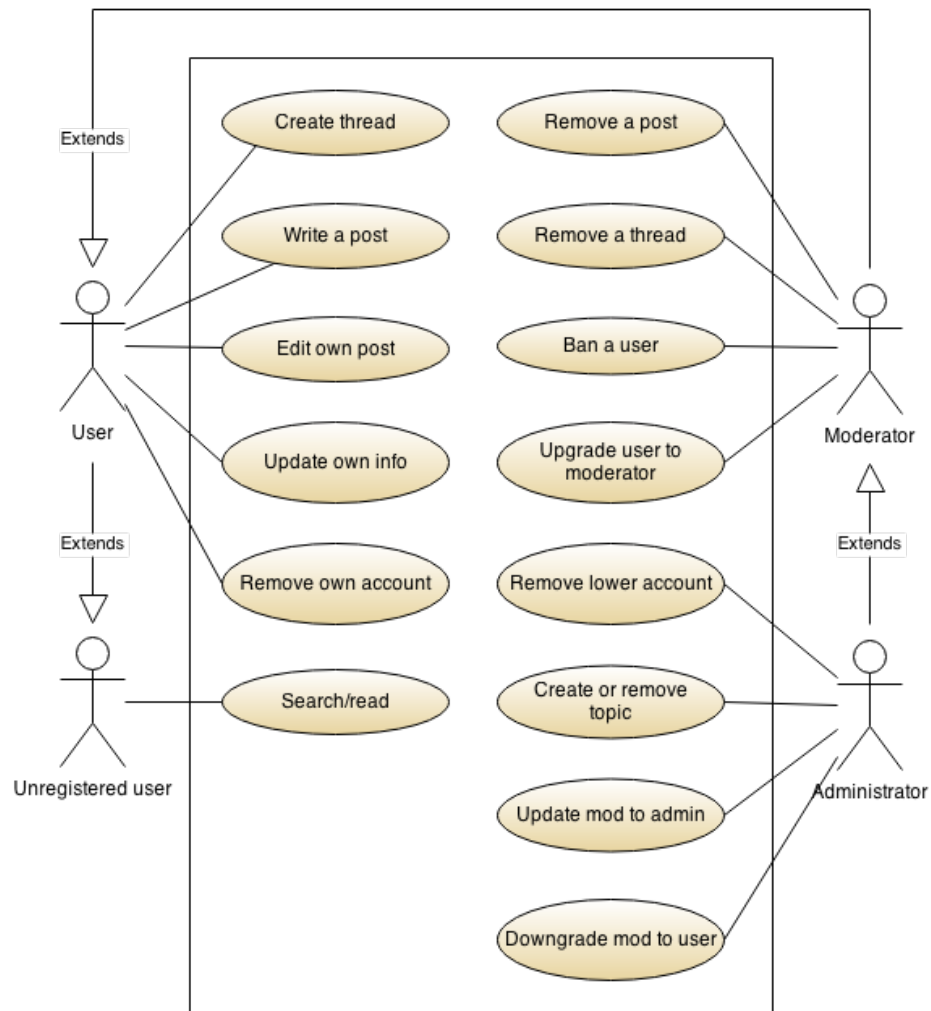
luo jokaiselle käyttäjälle profiilisivun, joissa näytetään (profiilin omistajan halutes-
sa) sähköpostiosoite, syntymäpäivä, sukupuoli, oikea nimi. Profiilisivuilla esitetään
aina (edes oletus-) avatar-kuva, lista postauksia ja postausten määrää.

Rekisteöityessään käyttäjä voi halutessaan lisätä avatar-kuvan, joka skaala-
taan tietynkokoiseksi; muussa tapauksessa käytetään oletus-avatar. Avatarkuvan
voi lisätä/poistaa ja tiedot päivittää myös rekisteröitymisen jälkeen.

multilog sallii viesteissä erityistagien kirjoittamisen, joilla tekstiä pystyy muo-
toilemaan; alustavasti seuraavat muotoilut toteutetaan:

- kursivointi,
- lihavointi,
- kiinteävälinen tekstitys (monospaced font),
- linkkien upottaminen,
- kuvien linkittäminen tekstiin.

2.1 Käyttötapaukset



Huomaa, että kaikki toiminnot paitsi lukeminen ja haku edellyttävät sisäänkirjautumisen.

2.2 Käyttäjärühmät

Jokamies

Henkilö joka ei ole rekisteröitynyt **multilog**:iin. Voi hakea ja lukea järjestelmässä olevat postaukset.

Peruskäyttäjä (user)

Rekisteröitynyt peruskäyttäjä, joka voi luoda säikeet ja postaukset, vastata

olemassaoleviin postauksiin, muokata tietojaan ja poistaa profilinsa. Perii myös **jokamiehen** toiminnot.

Moderattori (moderator)

Käyttäjä, joka voi poistaa säikeet ja postaukset, ylentää **peruskäyttäjän** moderaattoriksi ja aseta peruskäyttäjät käyttökieltoon. Perii myös **peruskäyttäjän** toiminnot.

Ylläpitäjä (admin)

Ylläpitäjä, joka voi ylentää peruskäyttäjän/moderaattorin ylläpitäjäksi, poistaa ei-ylläpitäjäkäyttäjän, luoda/poistaa aiheet. Perii myös **moderaattorin** toiminnot.

2.3 Käyttötapauskuvaukset

Jokamiehen käyttötapauskuvaukset

Haku:

Postausten, säikeiden, teemojen ja profiilien haku

Lukeminen:

Haetun sisällön lukeminen

Peruskäyttäjän käyttötapauskuvaukset

Säikeen luonti:

Peruskäyttäjä voi luoda säikeen teeman alle

Kirjoittaminen:

Peruskäyttäjä voi kirjoittaa säikeessä

Vastaaminen:

Peruskäyttäjä voi vastata olemassa olevaan viestiin

Muokkaaminen:

Peruskäyttäjä voi muokata omat viestit

Tietojen päivittäminen:

Peruskäyttäjä voi päivittää henkilötietojaan;
myös poistaamaan tilinsä

**** Peruskäyttäjä perii kaikki jokamiehen toiminnot ****

Moderaattorin käyttötapaukset

Säikeen poisto:

Moderaattori voi poistaa mielivaltaisen säikeen

Käyttökiellon asettaminen:

Moderaattori voi aseta mielivaltaisen peruskäyttäjän käyttökieltoon (maksimissaan 7 päiväksi)

Putsaaminen:

Moderaattori voi poistaa mielivaltaisen peruskäyttäjän viestin

Ylentäminen:

Moderaattori voi promotoida peruskäyttäjän moderaattoriksi

**** Moderattori perii kaikki peruskäyttään toiminnot ****

Ylläpitäjän käyttötapaukset

Y.1 Ylentäminen:

Ylläpitäjä voi promotoida moderaattorin ylläpitäjäksi

Y.2 Alentaminen:

Ylläpitäjä void alentaa moderaattorin peruskäyttäjäksi

Y.3 Käyttäjän poisto:

Ylläpitäjä voi poistaa peruskäyttäjän
(Y.2:n nojalla myös moderaattorin)

**** Ylläpitäjä perii kaikki moderaattorin toiminnot ****

3 Käyttöliittymä

Sovelluksen pääsivulla tulee olemaan 4 linkkiä:

Search sovelluksen hakutoiminto,

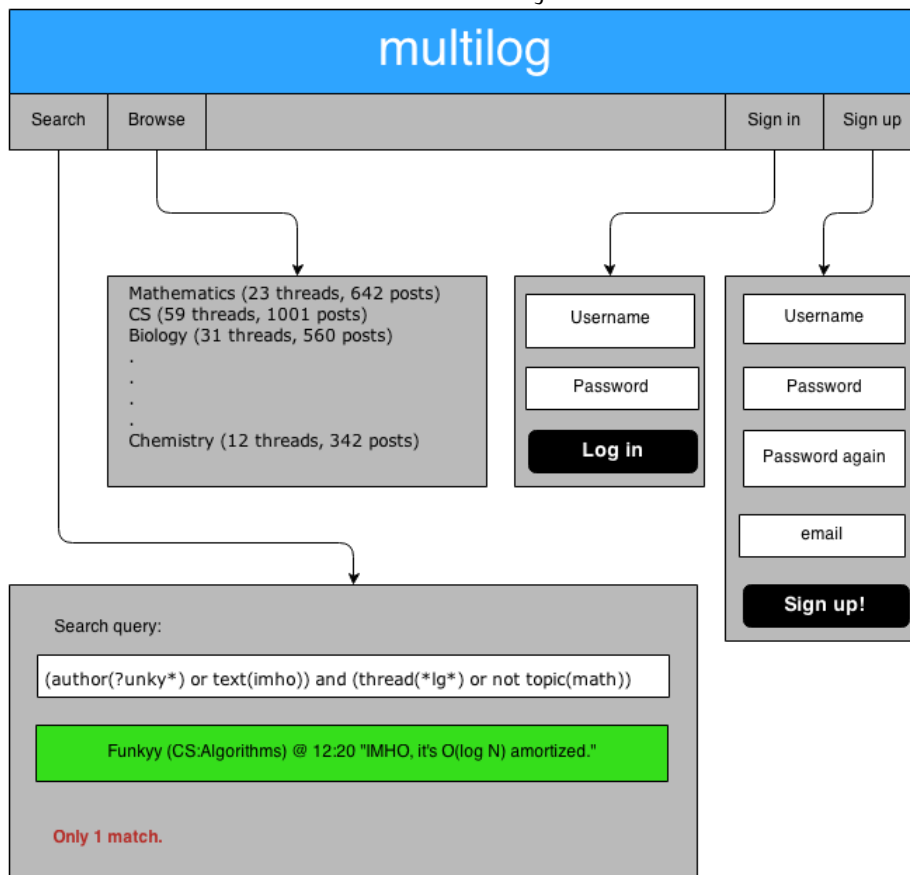
Browse ohjaa teemat sisältävälle sivulle, josta pääsee katsomaan kunkin teeman säikeet, joissa, puolestaan, pääsee lukemaan postaukset,

Sign in ohjaa sisäänkirjautumissivulle,

Sign up ohjaa rekisteröimissivulle.

Search - linkin takana on sivu, jossa käyttäjä voi kirjoittaa epätriviaaleja hakulausekkeita, joissa on operaattorit **NOT**, **AND** ja **OR**, joiden lisäksi on valitsimet *post*, *author*, *topic* ja *thread*. **Browse** - linkin takana on puolestaan lista teemoja, joita klikkaamalla pääsee listaan säikeitä, joita klikkaamalla pääsee lukemaan ko. säikeen postaukset (käyttäjän ollessa kirjautunut pääsee kirjoittamaan oman postauksen tai vastaamaan olemassaolevaan postaukseen). **Sign up** - linkin takana on sivu, jossa voi luoda peruskäyttäjän profiiliin ja **Sign in** - linkin takana järjestelmään voi kirjautua.

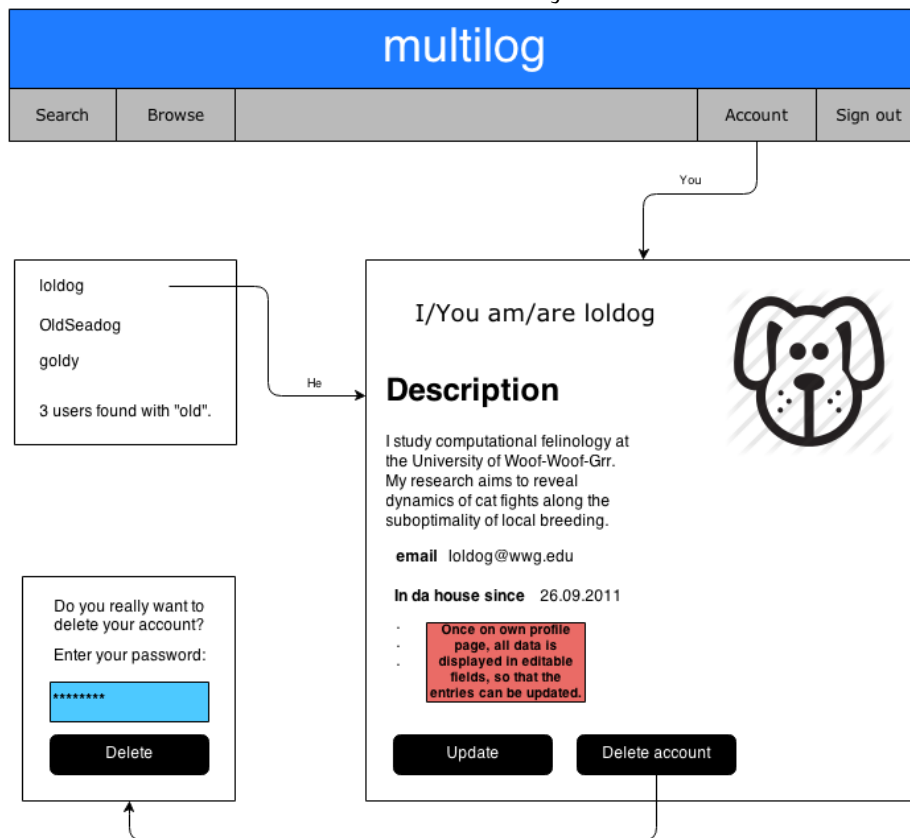
Kuva 1: Päänäkymä



Käyttäjän ollessa kirjautunut, navigointipalkin oikealla puolella on linkit - **Account** ja **Sign out**, joilla voi mennä oman tilin hallintasivulle tai kirjautua ulos vastaavasti. Jokainen *kirjautunut* käyttäjä K_1 voi lukea toisen käyttäjän K_2 tiedot sivulla S : lukijan K_1 ollessa korkeammalla käyttäjätasolla, voi hän promotoida K_2 :n enintään K_1 :n tasolle painamalla painiketta sivulla S .

Kirjautuneen käyttäjän selatessa säikeen viestit, voi hän painaa postauksen yhteydessä olevaa **Reply** - painiketta ja vastata siihen. Ellei käyttäjä halua nimenomaan vastata mihinkään postaukseen, voi hän kirjoittaa säikeen lopussa olevaan postauslaatikkoon, jolloin luodaan viesti vailla "vanhempiviestiä". Kuva 1 sivulla 6 demoaa **multilog**:n päänäkymää; kuva 2 (sivu 7) puolestaan demoaa profiilinäkymää ja, lopulta, 3 (sivu 8) näyttää suurinpiirtein sen, miltä säienäkymä tulee näyttää.

Kuva 2: Profiilinäkymä



Kuva 3: SÄitenäkymä

multilog				
Search	Browse		Account	Sign out
multilog >> Computer science >> Efficient heap data structures				
loldog @ 10.04.2011 12:56 wrote:			Reply	
Hello, all! I need an efficient priority queue as to facilitate the shortest path search. Any recommendations?				
goldy @ 10.04.2011 13:27 wrote:			Reply	
In case your graph contains, say, less then 1e4 nodes, it makes sense to use a binary heap. Otherwise, use Fibonacci heap from your favourite data structure library. :)				
<hr/>				
Forget the Fibonacci heap at once! It may be more efficient in terms of running time bounds, yet it's hidden constant factors are huuu				
Showing all 2 posts.				Post