

Muistilista

Tietokantasovelluksen harjoitustyö
Tekijä: Mikael Wide
OpNro: 014248926
mikael.wide@helsinki.fi
16.3.2014

Sisällysluettelo

1. Johdanto.....	3
2. Järjestelmän kuvaus.....	3
2.1 Käyttäjät.....	3
2.2 Käyttötapaukset.....	4
2.2.1 Käyttötapauskaavio.....	4
2.2.2 Peruskäyttäjän käyttötapaukset.....	4
3. Järjestelmän tietosisältö.....	5
3.1 Tietokohteet.....	6
3.1.1 Tietokohde: Users.....	6
3.1.2 Tietokohde: Luokka.....	6
3.1.3 Tietokohde: Tärkeysaste.....	6
3.1.4 Tietokohde: Askare.....	7
4. Relaatiotietokantakaavio.....	8
5. Järjestelmän yleisrakenne.....	8
6. Käyttöliittymä.....	9
7. Asennustiedot.....	10
8. Käyttöohje.....	10
9. Testaus, tunnetut bugit ja puutteet & jatkokehitysideat.....	10
10. Omat kokemukset.....	11

1. Johdanto

Tietokantasovellus Muistilista on web-sovellus, joka käyttää apunaan PostgreSQL-tietokantaa. Muistilista on apuväline päivittäisten askareiden muistamiseen. Sovellukseen voidaan tallentaa päivittäisiä askareita, joille voi antaa tietyn tärkeysasteen ja sijoittaa luokkiin. Askareita voi sitten katsoa listassa jossa ne ovat järjestetty tärkeysjärjestyksen mukaan.

Sovellus toteutetaan PHP-llä joka käyttää tietojen tallentamiseen PostgreSQL-relaatiotietokantaa. Sovellus toteutetaan Helsingin Yliopiston Tietojenkäsittelytieteen laitoksen users-palvelimella, osoitteessa mikaewi.users.cs.helsinki.fi/Tsoha. Sovelluksen käyttäjältä ei vaadita muuta kuin pääsy internetiin jonkun nykyaikaisen web-selaimen kautta.

2. Järjestelmän kuvaus

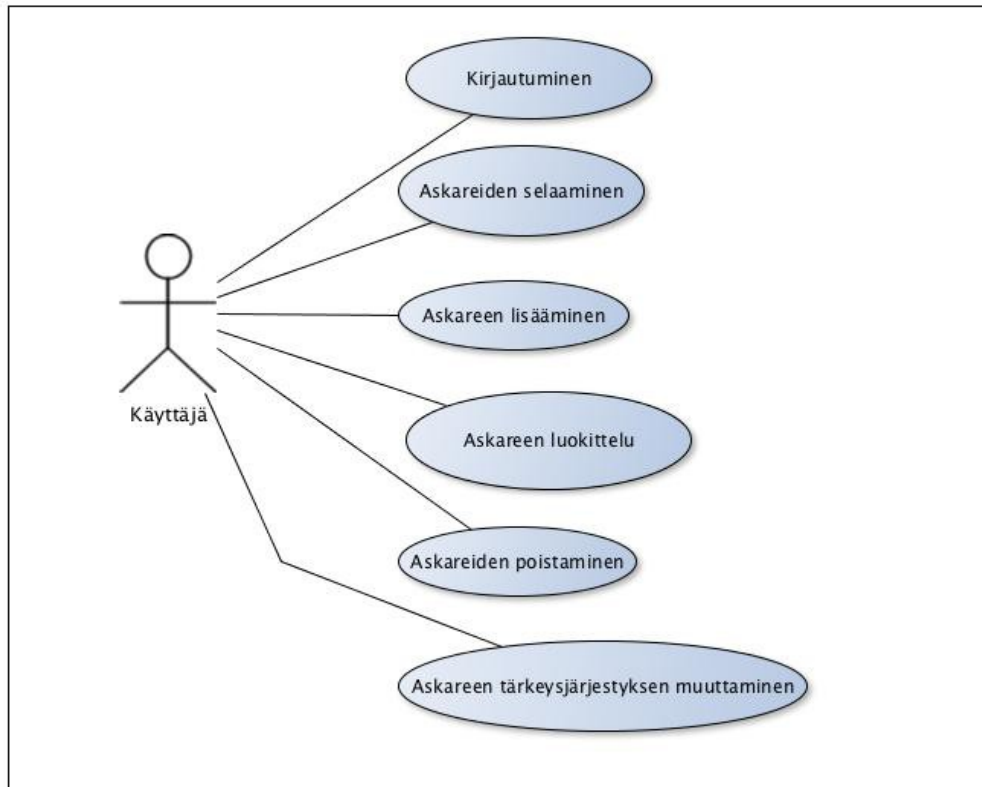
2.1 Käyttäjät

Peruskäyttäjä – Käyttäjä joka voi kirjautua saamallaan tunnuksella ja salasanalla, tai rekisteröidä oman tunnuksen. Käyttäjä voi sitten lisätä uusia askareita, määrittää niille luokkia ja tärkeysjärjestyksiä. Askareita voi selata jälkeensä loogisessa järjestyksessä prioriteettien avulla, siten että tekemättömät askaret ovat listassa ensin jonka jälkeen katsotaan prioriteettia. Käyttäjällä on myös mahdollista lisätä ja poistaa luokkia ja prioriteetteja jälkikäteen.

Ylläpitäjä- Ylläpitäjällä on muuten kaikki samat toiminnot käytössä kuin peruskäyttäjällä mutta pystyy myös poistamaan muita käyttäjiä. Ylläpitäjän tiliä ei voida poistaa.

2.2 Käyttötapaukset

2.2.1 Käyttötapauskaavio



2.2.2 Peruskäyttäjän käyttötapaukset

Rekisteröityminen:

Käyttäjä voi ensin rekisteröityä haluamallaan tunnuksella ja salasanalla.

Kirjautuminen:

Peruskäyttäjä kirjautuu saamallaan tunnuksella ja salasanalla järjestelmään.

Askareiden lisääminen:

Peruskäyttäjä lisää uuden askareen ja määrittää sille samalla tärkeysasteen. Askareen voi myös määrittää kuulumaan tiettyyn luokkaan mutta se ei ole pakollista.

Askareen luokittelu:

Valmiiksi luotu askare voidaan määrittää kuulumaan tiettyyn luokkaan.

Askareen tärkeysjärjestyksen muuttaminen:

Valmiiksi luodun askareen tärkeysjärjestystä muutetaan.

Askareiden selaaminen:

Perukäyttäjä voi selata askareita listassa joka on järjestetty loogisesti tärkeysjärjestyksen mukaan.

Askareiden poistaminen:

Peruskäyttäjä voi poistaa haluamansa askareet.

Luokkien selaaminen/poistaminen:

Käyttäjä voi selata ja halutessaan poistaa jonkun luokan. Askare johon luokka on kiinnitetty ei kuitenkaan poistu vaikka sen luokka poistetaan.

Luokkien lisääminen:

Käyttäjä voi lisätä luokkia ja määrittää niille ylliluokan jos haluaa.

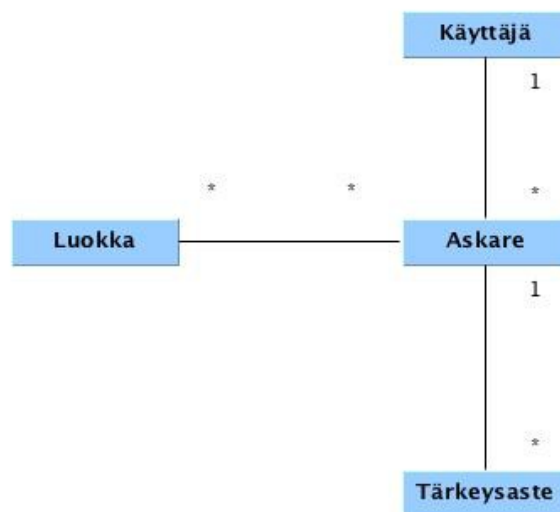
Tärkeysasteiden selaaminen/poistaminen/lisääminen.

Käyttäjä voi lisätä ja poistaa tärkeysasteita.

Askareen tehdyksi merkitseminen:

Käyttäjä voi merkitä luomansa askareen tehdyksi klikkaamalla 'Tehty'-sarakkeessa olevaa nappulaa sen askareen kohdalta.

3. Järjestelmän tietosisältö



Käsitekaavio

3.1 Tietokohteet

3.1.1 Tietokohde: Users

Atribuutti	Arvojoukko	Kuvailu
id	Serial	Taulun automaattisesti generoitu pääavain
kayttajanimi	Varchar	Käyttäjänimi
salasana	Varchar	Salasana

Users-tauluun tallennetaan käyttäjiä ja niihin liittyvät salasanat.

3.1.2 Tietokohde: Luokka

Atribuutti	Arvojoukko	Kuvailu
id	Serial	Taulun automaattisesti generoitu pääavain
otsikko	Varchar	Luokan nimi, esim: Kotityöt
kuvaus	Text	Luokan kuvaus
ylaluokka_id	Integer	Luokalle kuuluva Yläluokka jos määritetty

Luokalla on otsikko, kuvaus ja viite yläluokkaan, joka siis voi viitata itseensä jos niin halutaan. Luokka voi kuulua moneen askareeseen

3.1.3 Tietokohde: Tärkeysaste

Atribuutti	Arvojoukko	Kuvailu
id	Serial	Taulun automaattisesti generoitu pääavain
otsikko	Varchar	Tärkeysasteen otsikko, esim: Erittäin tärkeä
prioriteetti	Integer	Tärkeysasteen prioriteetti lukuna.

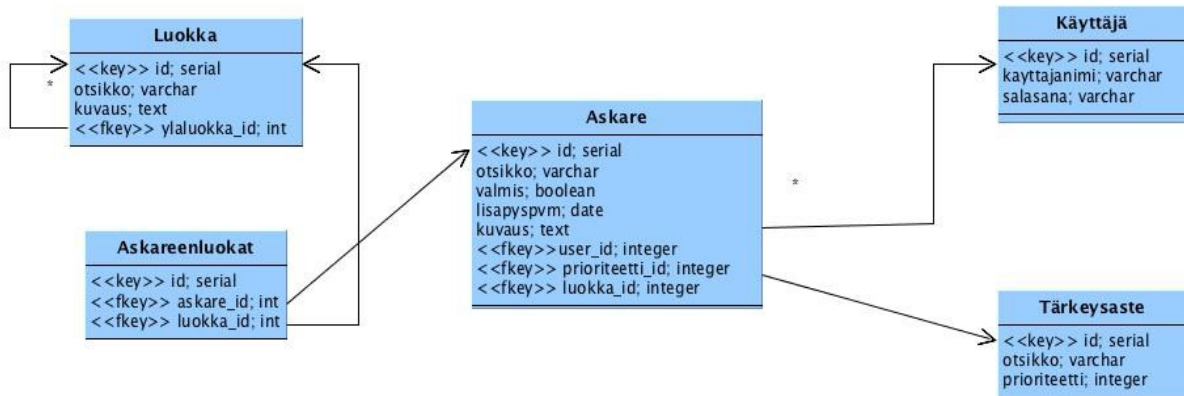
Tärkeysaste-tauluun tallennetaan tärkeysasteet ja otsikot. Jokaiselle tärkeysasteelle annetaan jokin prioriteetti kokonaislukuna. Esim prioriteetteja 1-6 jossa isompi luku tarkoittaa korkeampaa tärkeysastetta. Tärkeysaste voi kuulua moneen askareeseen mutta yhdellä askareella voi vain olla yksi tärkeysaste.

3.1.4 Tietokohde: Askare

Atribuutti	Arvojoukko	Kuvailu
id	Serial	Taulun automaattisesti generoitu pääavain
otsikko	Varchar	Askareen nimi, esim: Tiskit
valmis	Boolean	True jos askare on tehty, muuten ja oletuksena false.
lisayspvm	Date	Askareen lisäyspäivämäärä.
kuvaus	Text	Askareen kuvaus

Askareeseen liitetään viiteavaimena aina jokin käyttäjä. Siihen voi myös liittää tärkeysasteen ja/tai luokan.

4. Relaatiotietokantakaavio

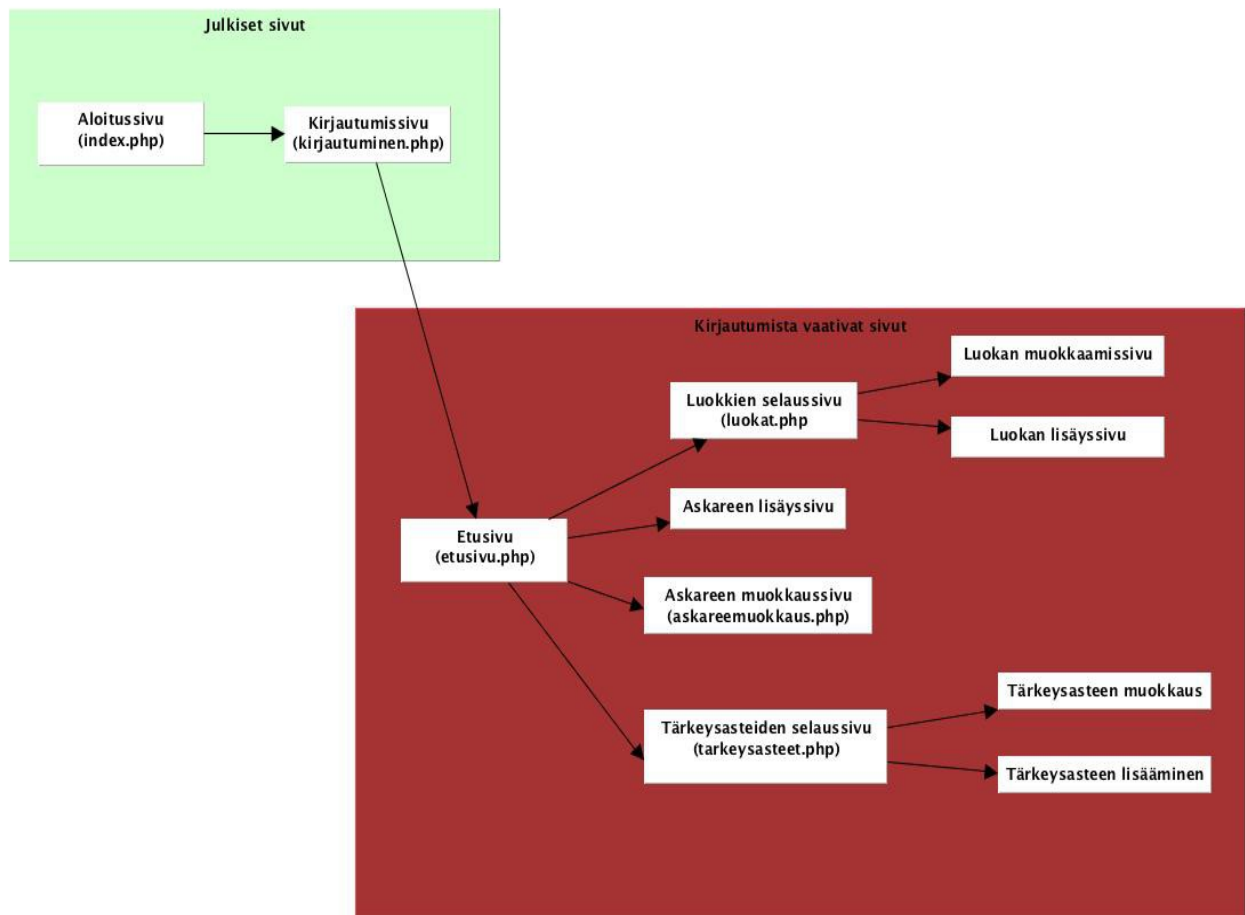


5. Järjestelmän yleisrakenne

Sovelluksessa on noudatettu MVC-mallin mukaista rakennetta. Mallit löytyvät kansiota `/libs/models`, näkymät kansiota `../views/` ja kontrollerit juurihakemistosta. Sovelluksessa on käytetty Twitter-Bootstrap-kirjaston valmiita tyylejä.

Kaikki tiedostonimet on pienellä, poislukien kaikki `/libs/models/` kansiota löytyvät luokat jotka ovat isolla alkukirjaimella.

6. Käyttöliittymä



7. Asennustiedot

Asenna sovellus kopioimalla sen tiedostot palvelimen nettiin näkyvään hakemistoon (esim. usersin htdocs-hakemisto). Aseta sen jälkeen tietokannan yhteystiedot oikeaksi tiedostoon `libs/common.php` metodiin `getTietokantayhteys()`. PostgreSQL-kannan saa toimimaan ajamalla tiedoston `sql/create-tables.sql`. Sovellus käyttää yhtä dummy-riviä talusta tärkeysaste (`id=1`) eli syötä kannalle `sql/add-test-data.sql` tai lisää tauluun joku yksi rivi jolla merkitään sitä tilaa jolloin tietylle askareelle ei ole asetettu mitään tärkeysastetta, tai tärkeysaste on poistettu askareen luomisen jälkeen.

Sovelluksen käyttö ei ole riippuvainen asennussijainnista muuten kuin navigointipalkissa sijaitsevien linkkien kohdalla (`views/navi.php`). `Navi.php:ssä` linkit ovat absoluuttisia joten joudut muuttamaan niitä jos haluat käyttää sovellusta jossain muualla kuin sen nykyisessä sijainnissa.

8. Käyttöohje

Harjoitustyö on esillä osoitteessa: <http://mikaewi.users.cs.helsinki.fi/Tsoha/index.php>. Pääset kirjautumaan ylläpitäjän tilille tunnuksella: admin ja salasanalla admin. Tavalliseksi käyttäjäksi voi rekisteröityä ja käyttää tunnusta: tester ja salasanaa password. Ylläpitäjän tunnuksilla on muuten samat toiminnot kuin peruskäyttäjällä mutta pystyy myös poistamaan käyttäjiä kannasta.

9. Testaus, tunnetut bugit ja puutteet & jatkokehitysideat

Olen testannut ohjelmaa koko kehitysvaiheen ajan kokeilemalla kaikkia uusia toimintoja mahdollisimman monipuolisesti. Testaus on kuitenkin suoritettu täysin käsin eikä millään apuohjelmilla.

Ohjelmasta puuttuu esim käyttäjätietojen muokkaamisominaisuus, tärkeysasteiden ja luokkien muokkaus. Siinä voisi myös olla mahdollisuus järjestää askareita uudestaan niiden sarakkeiden avulla kuten haluaa.

Yksi iso vika joka ohjelmassa on tällä hetkellä, on se että kaikki luokat ja tärkeysasteet ovat kaikille käyttäjille yhteisiä. Vaikka kaikilla on omat askareet käytössä niin niihin liittyvät luokat ja tärkeysasteet ovat yhteisiä, joka on kieltämättä vähän hölmöä. Nyt jos yksi käyttäjä poistaa luokan, se häviää myös kaikilta muilta. Absoluuttiset linkit navigointipalkissa ovat myös hieman haitallisia jos haluaa siirtää ohjelman johonkin muualle pyörimään.

Kaikki ilmoitukset jotka ohjelma antaa tulevat todella rumasti sivun ylälaitaan mutten saanut niitä sopimaan mihinkään paremmin joten se oli käytännöllisin paikka pitää niitä.

10. Omat kokemukset

Alussa kaikki koko kurssiin liittyvä tuntui todella hankalta kun kaikki lukuun ottamatta pieni osa sql:ää oli uutta. HTML ja muu ulkonäköön liittyvä on vieläkin hieman hämärän peitossa mutta olen melko tyytyväinen siihen miten sain ohjelman käyttöliittymän näyttämään. MVC-mallin ymmärtäminen oli melko helppoa mutta alussa sen toteuttaminen php:n avulla tuntui hankalta.

Mielestäni onnistuin kuitenkin melko hyvin noudattamaan MVC:tä suuressa osassa sivuja, joskin välillä tuli hieman fuskattua. Tunnen että opin käyttämään PHP:n perussyntaksia ja myös ymmärtämään jotain siitä. Opin myös paljon enemmän web-sovelluksen rakenteesta koska suurin osa tekemästäni oli täysin uutta.

Jälkikäteen mietittynä olisi pitänyt käydä ohjauksissa kysymässä enemmän apua koska tuli tuskailtua kotona aivan liikaa joissain kohdissa.