

Tietokantojen harjoitustyö

Dokumentaatio 17.10.2014

Lassi Haasio  
014053438

## Sisällysluettelo

1. Johdanto
2. Yleiskuva järjestelmästä
3. Järjestelmän tietosisältö
4. Relaatiotietokantakaavio
5. Järjestelmän yleisrakenne
6. Käyttöliittymä
7. Asennustiedot
8. Käynnistys- /käyttöohje
9. Testaus, tunnetut bugit ja puutteet & jatkokehitysideat
10. Omat kokemukset

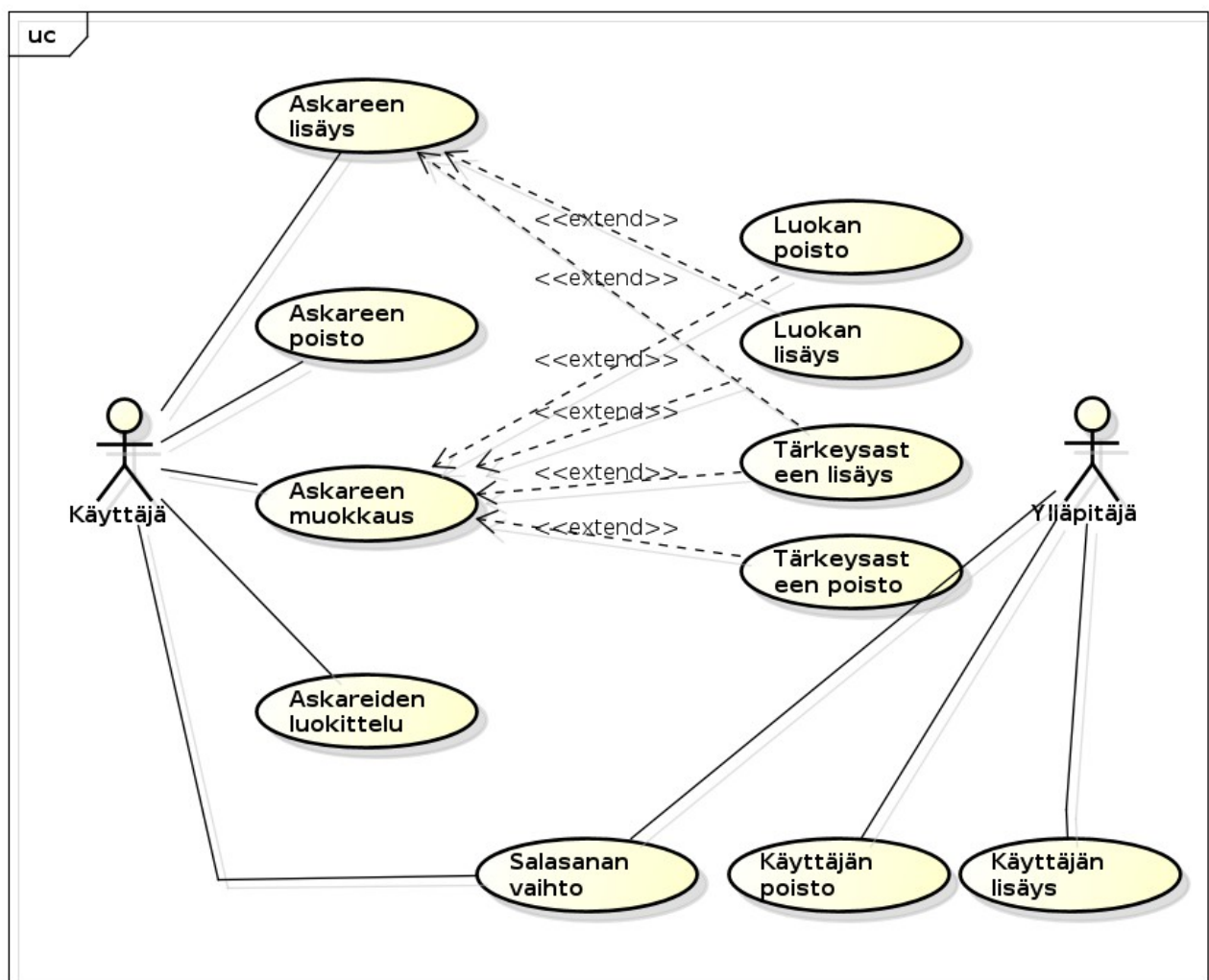
## 1. Johdanto

Harjoitustyön aihe on muistilista. Muistilista sijaitsee palvelimella, jolloin useat eri käyttäjät voivat käyttää omaa listaansa internetin välityksellä. Tarkoituksena on, että järjestelmään voi lisätä omia askareitaan muistiin niin että askareet ovat tärkeysjärjestyksessä. Askareita voi myös luokitella erilaisiin luokkiin ja luokat voivat edelleen jakautua aliluokkiin. Käyttäjä voi lisätä haluamiaan luokkia tahtonsa mukaan. Tarvittaessa käyttäjä voi myös muokata jo tallennettua askareta. Askareen tärkeyttä voi muuttaa, ja siihen voi lisätä tai siitä voi poistaa luokkia.

Työ toteutetaan web-pohjaisena sovelluksena koulun users-palvelimella PHP-kielellä. Tietokantana käytetään PostgreSQL-tietokantaa.

## 2. Yleiskuva järjestelmästä

### Käyttötapauskaavio



powered by Astah

Kaaviosta puuttuu käyttötapaus ”kirjautuminen”, sillä se sisältyy kaikkiin käyttötapauksiin ja se sotkisi käyttötapauskaavion.

### Käyttäjäryhmät

Käyttäjä:

Järjestelmän tavallinen käyttäjä, joka käyttää järjestelmää muistilistan ylläpitoon.

Ylläpitäjä:

Henkilö, joka hallinnoi käyttäjien tilejä ja tarvittaessa perustaa uusia ja poistaa vanhoja.

### Käyttäjän käyttötapaukset

Askareen lisäys

Käyttäjä lisää muistilistaansa askareen ja antaa sille nimen. Käyttäjä voi myös luokitella askareen tai antaa sille tärkeysasteen.

Luokan lisäys

Käyttäjä lisää uuden luokan lisätessään uutta askareta tai muokatessaan vanhaa.

Luokan poisto

Käyttäjä poistaa luokan. Luokassa olevilta askareilta poistetaan kyseinen luokka.

Askareiden luokittelu

Käyttäjälle näytetään kaikki luokat, ja hän valitsee minkä luokan askareet tahtoo nähdä.

### Ylläpitäjän käyttötapaukset

Käyttäjän lisäys

Ylläpitäjä lisää järjestelmään uuden käyttäjän. Ylläpitäjä valitsee käyttäjän käyttäjätunnuksen ja salasanan.

Muita käyttötapauksia: Kirjautuminen, askareen poisto, askareen muokkaus, tärkeysasteen lisäys, tärkeysasteen poisto, käyttäjän poisto, salasanan vaihto

## 3. Järjestelmän tietosisältö

### Tietokohde: Käyttäjä

Attribuutti	Arvojoukko	Kuvailu
Käyttäjätunnus	Merkkijono, pituus rajoittamaton	
Salasana	Merkkijono, pituus rajoittamaton	
Admin	Totuusarvo	TRUE jos käyttäjä on ylläpitäjä

### Tietokohde: Askare

Attribuutti	Arvojoukko	Kuvailu
kayttaja_id	Kokonaisluku	Vierasavain käyttäjät-tauluun
Kuvaus	Merkkijono, pituus rajoittamaton	Käyttäjän antama kuvaus askareelleen
Tärkeys	Kokonaiss[1,5]	Askareen tärkeysaste, 1 on tärkein

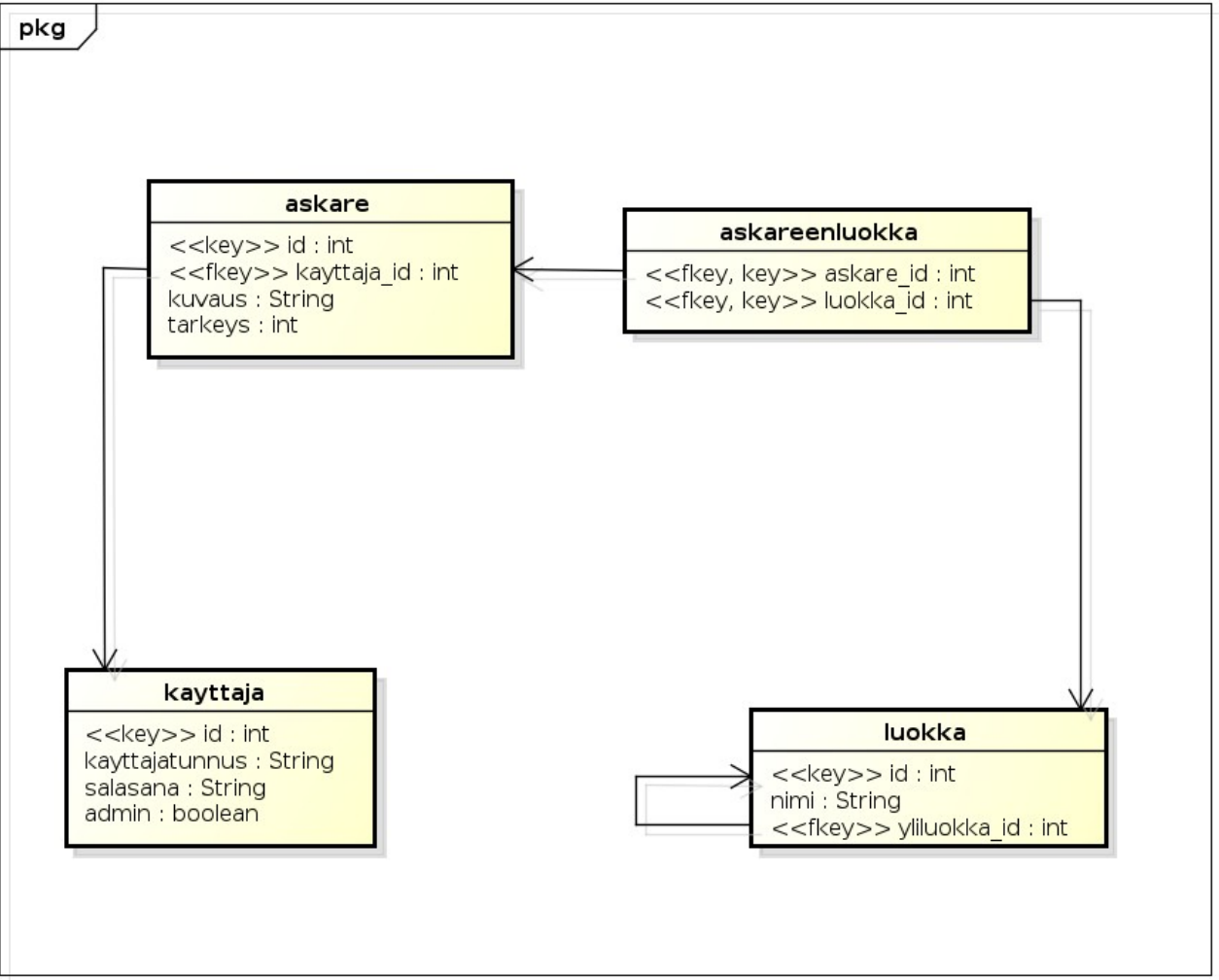
### Tietokohde: Luokka

Attribuutti	Arvojoukko	Kuvailu
Nimi	Merkkijono, pituus rajoittamaton	
yliluokka_id	Kokonaisluku	Vierasavain mahdolliseen yliluokkaan

### Tietokohde: Askareenluokka

Monesta moneen -suhteen aputaulu askareen ja luokan välille.

#### 4. Relaatiotietokantakaavio

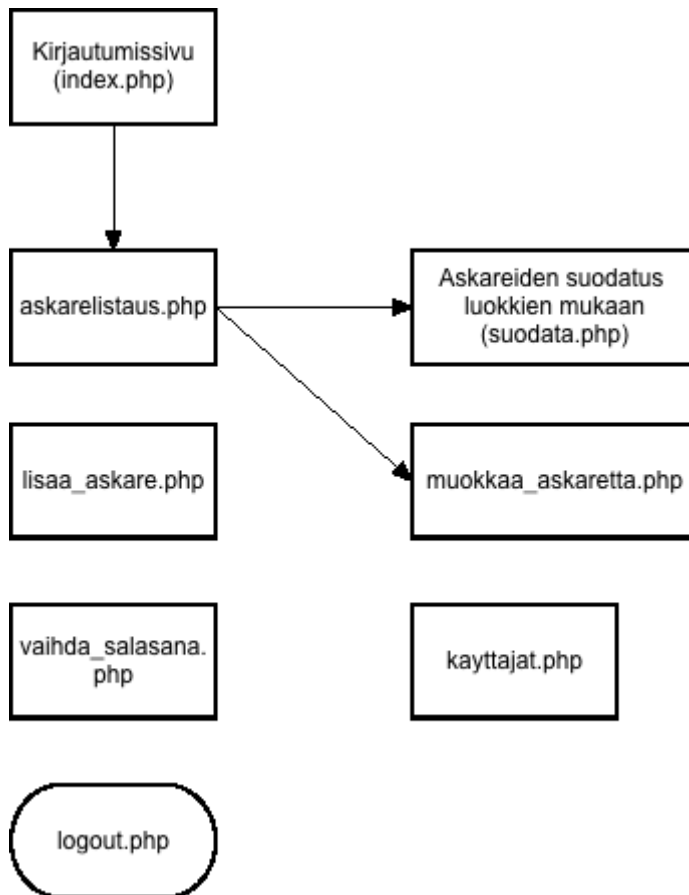


## 5. Järjestelmän yleisrakenne

Sovellus on toteutettu MVC-mallin mukaisesti. Kontrollerit ovat repositorion juuritasossa, näkymät hakemistossa /views ja mallit hakemistossa /lib/models. Apuna käytetty kirjasto on hakemistossa /lib. Kontrollerit on nimetty samalla nimellä, kuin niiden näkymän esittävät näkymätiedostot. Poikkeuksena on kontrolleri index.php, joka toimii kirjautumisen kontrollerina. Nimeämisen syynä on se, että sovellus toimisi suoraan palvelimen URL-osoitteen avulla.

Järjestelmä käyttää istuntoa tallentamalla \$\_SESSION-muuttujaan kirjautuneen käyttäjän id-numeron. Tämän avulla tietokannasta haetaan käyttäjän omat askareet ja muu käyttäjälle suunnattu sisältö.

## 6. Käyttöliittymä



Sivut askarelistaus.php, lisaa\_askare.php, vaihda\_salasana.php sekä logout.php ovat navigointipalkissa, minkä vuoksi niihin ei ole piirretty nuolia. Ylläpitäjällä navigointipalkissa on lisäksi kayttajat.php. Ainoastaan kirjautumissivu on julkinen. Kaikkia muita sivuja pääsee käyttämään vain kirjautunut käyttäjä.

## 7. Asennustiedot

Asenna sovellus kopioimalla sen tiedostot palvelimen nettiin näkyvään hakemistoon (esimerkiksi usersin htdocs-hakemisto). Aja sen jälkeen sql-kansiossa oleva `create-tables.sql` -tiedosto.

## 8. Käynnistys- /käyttöohje

Esittelysivu:

<http://lhaasio.users.cs.helsinki.fi/esittelysivu.html>

Sovelluksen kirjautumissivu:

<http://lhaasio.users.cs.helsinki.fi>

Pääkäyttäjän tunnukset:

Käyttäjätunnus: admin

Salasana: kermanekka



## 9. Testaus, tunnetut bugit ja puutteet & jatkokehitysideat

Testaus on toteutettu lähinnä kokeilemalla tehdä askarelistalla kaikkea mahdollista puuttuvista ja virheellisistä syötteistä aina GET-parametrien peukalointiin. Mielestäni näin pienessä sovelluksessa moinen testaus on riittävää.

Olen projektin aikana kerännyt listaa bugeista ja puuttuvista ominaisuuksista TODO.txt-tiedostoon. Kaikki paitsi numero 8 (Luokkaa lisättäessä uusi luokka ei näy askarelistauksessa, mikäli lomakkeen syöte on virheellinen) on korjattu. En onnistunut toistamaan bugia 8, joten se jäänee työn tarkastajan löydettäväksi. Muita tunnettuja bugeja ei ohjelmassa ole.

Näin jälkikäteen kannattaisi adminille lisätä mahdollisuus nollata käyttäjän salasana unohtamisen varalta. Käyttäjien poistoon ja oman admin-statusen poistoon pitäisi lisätä ”Oletko aivan varma?” -kysymys. Nyt käyttäjän voi poistaa helposti vahingossa ilman mahdollisuutta perua tekoa. Luokkien lisäys ja käsittely voisi olla helpompaa. Nyt luokkia voi lisätä vain yksi kerrallaan. Luokalle voi myös asettaa yliluokan vain luontihetkellä.

## 10. Omat kokemukset

Kurssilla oli todella hyvä ohjeistus. Kun sen päälle oli pajamuotoista ohjausta, ei mitään tarvinnut jäädä arpomaan. Ihan toista kuin matematiikan ja tilastotieteen laitoksen opetus. Työtä sen sijaan oli todella paljon. Toisaalta koen oppineeni todella paljon kurssilla, ja positiivista oli, että lopputulos on oikean elämän sovellus, eikä hypoteettinen malli, kuten monella muulla kurssilla.

Petyin siihen, että sain vasta demotilaisuudessa tietää, ettei sovelluksia oikeasti suunnitella MVC-arkkitehtuurilla. Demotilaisuudessa kerrottiin, että MVC-malli on vanhentunut, ja tekee yhtään isompien sivustojen toteuttamisesta hankalaa. Ei ole hyväksi opettaa opiskelijoille vanhentuneita menetelmiä. Koko yliopiston filosofinen ero ammattikorkeakouluun on se, että oman tutkimuksen tähden opetuksessa voidaan hyödyntää uusinta ja edistyneintä tietoa. Tämä kurssi ei oikein muistanut olevansa yliopiston kurssi.

Assistentille erityiskiitos. Ymmärsit kysymykset hyvin, ja tiesit aina oikean vastauksen. Moista luksusta ei yleensä kursseilta voi olettaa.