

MUISTILISTA

Tuisku Haro

Helsingin yliopisto

Tietojenkäsittelytieteen laitos

Tietokantasovellus

Loppukesä 2015

SISÄLLYSLUETTELO

1. Johdanto

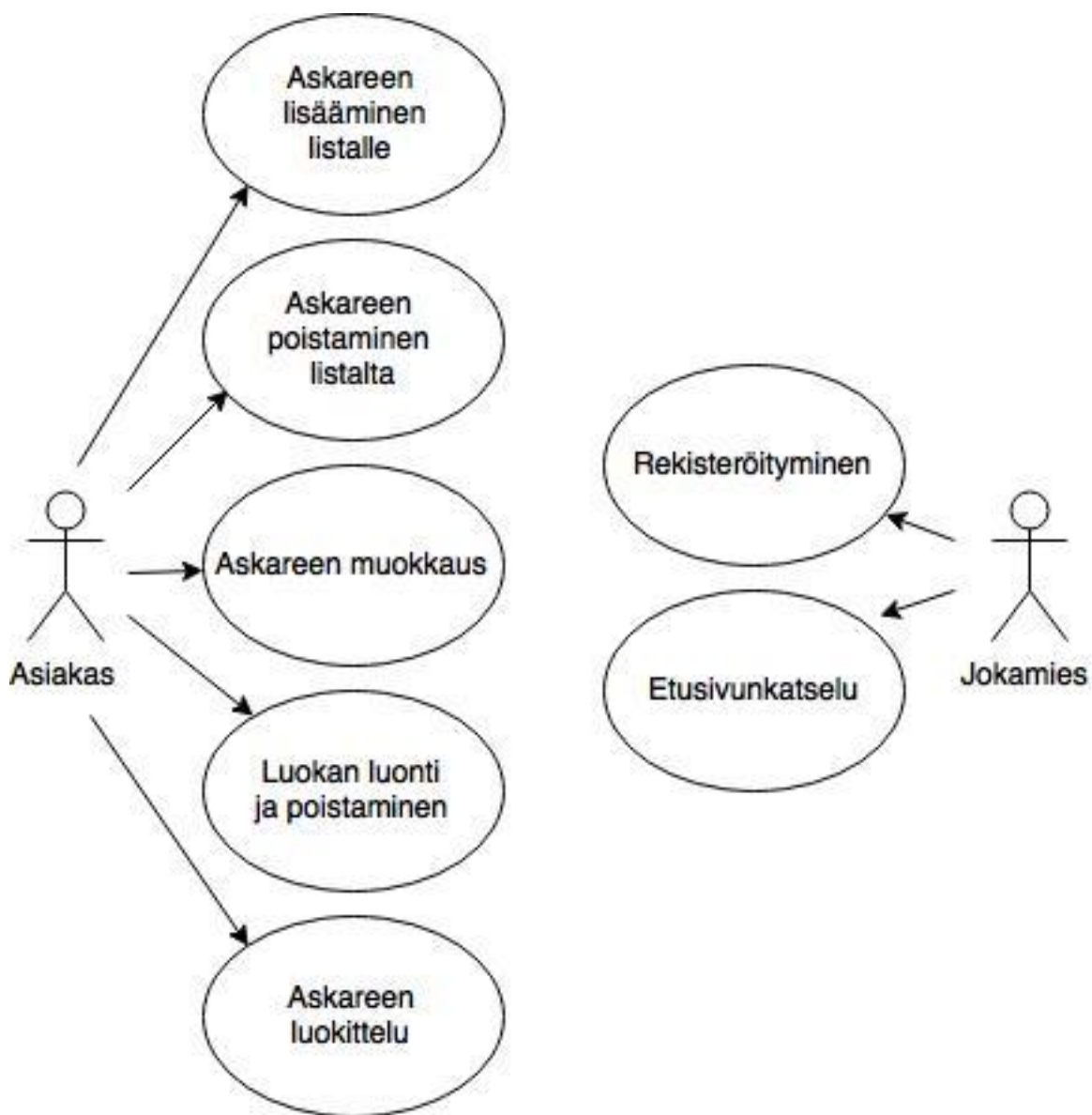
Valitsin tietokantasovellukseni aiheeksi palvelun muistilistan tekemistä varten. Muistilistaan pitäisi pystyä tallettamaan päivän eri askareita, jotka olisivat järjestyksessä listalla askareen tärkeyden mukaan. Askareelle pitäisi siis itse pystyä asettamaan oman tärkeysasteen, joko askareen luonnin yhteydessä tai myöhemmin erikseen. Tämän lisäksi askareet kuuluisi pystyä luokittelemaan erilaisiin ryhmiin, eli tarvittaisiin ominaisuudet luokkien luomiselle ja poistamiselle, sekä askareiden luokittelumahdollisuudet. Askareen pitäisi pystyä luokittelemaan useampaan, kuin vain yhteen luokkaan ja luokkien kuuluisi voida olla sisäkkäisiä, jottei luokittelu jäisi liian tönköksi.

Palvelu on myös tarkoitettu useamman ihmisen käytettäväksi, eli jokaisella käyttäjälle tarvitaan omat tunnukset, joihin kuuluu myös omat muistilistat ja niiden tärkeysasteet ja luokat. Tätä varten tarvittaisiin myös tietenkin kirjautumismahdollisuus.

Käytän Javaa ja PostgreSQL:ää palvelu toteuttamiseksi. Selaimen alustajärjestelmän pitää siis tukea ainakin Javaa. Ympäristönä aion käyttää laitoksen users-palvelinta. En usko, että tietokannan vaihdon jälkeen palvelu toimisi enää.

2. Yleiskuva järjestelmästä

Käyttötapauskaavio:



Käyttäjärühmät:

Jokamiehellä tarkoitetaan ketä tahansa joka on päätenyt Muistilista palvelun sivuille.

Asiakas on palvelun rekisteröitynyt käyttäjä.

Käyttötapauskuvaukset:

Jokamiehen käyttötapaukset:

Etusivunkatselu: Jokamies pystyy lukea palvelun ominaisuuksista etusivulta.

Muita käyttötapauksia: rekisteröityminen, kirjautuminen.

Asiakkaan käyttötapaukset

Askareen lisääminen listalle: Jokaisella asiakkaalla on oma listansa, johon he pystyvät lisäämään askareita.

Esimerkki: Asiakas lisää askareen imurointi listalle tärkeysasteella 3.

Askareen poistaminen listalta: Asiakas voi halutessaan poistaa askareita listaltansa.

Esimerkki: Asiakas on suorittanut imuroimisen ja haluaa täten ottaa sen pois listalta.

Askareen muokkaus: Asiakas voi halutessaan muuttaa aiemmin lisäämänsä askareen tietoja.

Esimerkki: Asiakas muuttaa askareen ”pese ikkunat” tärkeysastetta

Luokan luonti ja poistaminen: Asiakas pystyy halutessaan luoda ja poistaa luokkia.

Esimerkki: Asiakas luo luokan ”pihatyöt”.

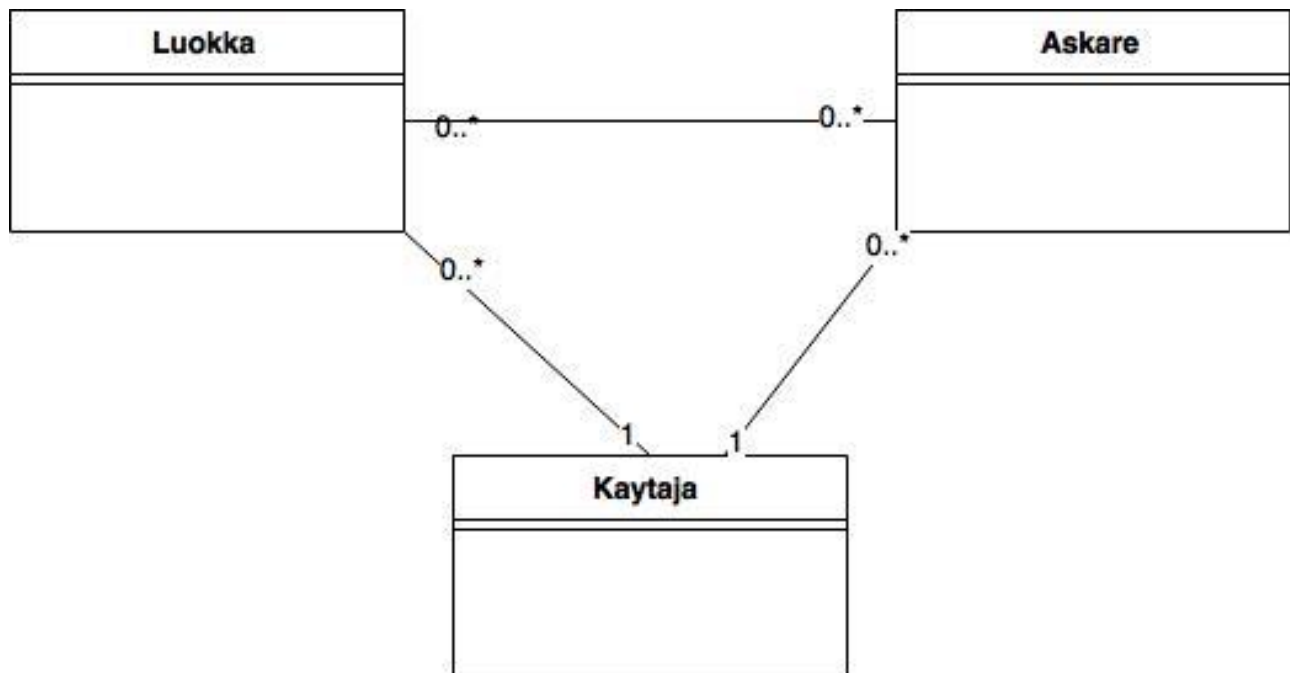
Askareen luokittelu: Asiakas voi luokitella askareita erilaisiin luokkiin, sekä poistaa askareita luokista.

Esimerkki: Asiakas lisää askareen ”aja nurmikko” luokkaan ”pihatyöt”.

Muita käyttötapauksia: kirjautuminen.

3. Järjestelmän tietosisältö

3.1 Käsitekaavio:



3.2 Tietokohteiden määrittely:

Tietokohde: Askare

Atribuutti	Arvojoukko	Selitys
askareenId	SERIAL	Automaattisesti generoitu tunnus
tarkeysarvo	varchar(3)	Käyttäjän asettama tarkeysarvo askareelle
nimi	varchar(40)	Käyttäjän asettama nimi askareelle
kayttajaid	integer	fkey kayttajaan
luokkaid	integer	fkey luokkaan

Askare taulussa on jokaisen yksittäisen askareen tiedot, sekä viittaukset toisiin tauluihin ja uniikki id askareelle. Askare voi kuulua moneen eri luokkaan. Askareella on vain yksi käyttäjä.

Tietokohde: Kayttaja

Atribuutti	Arvojoukko	Selitys
kayttajaid	SERIAL	Automaattisesti generoitu tunnus
kayttajatunnus	varchar(10)	Käyttäjän asettama käyttäjätunnus
salasana	varchar(20)	Käyttäjän asettama salasana

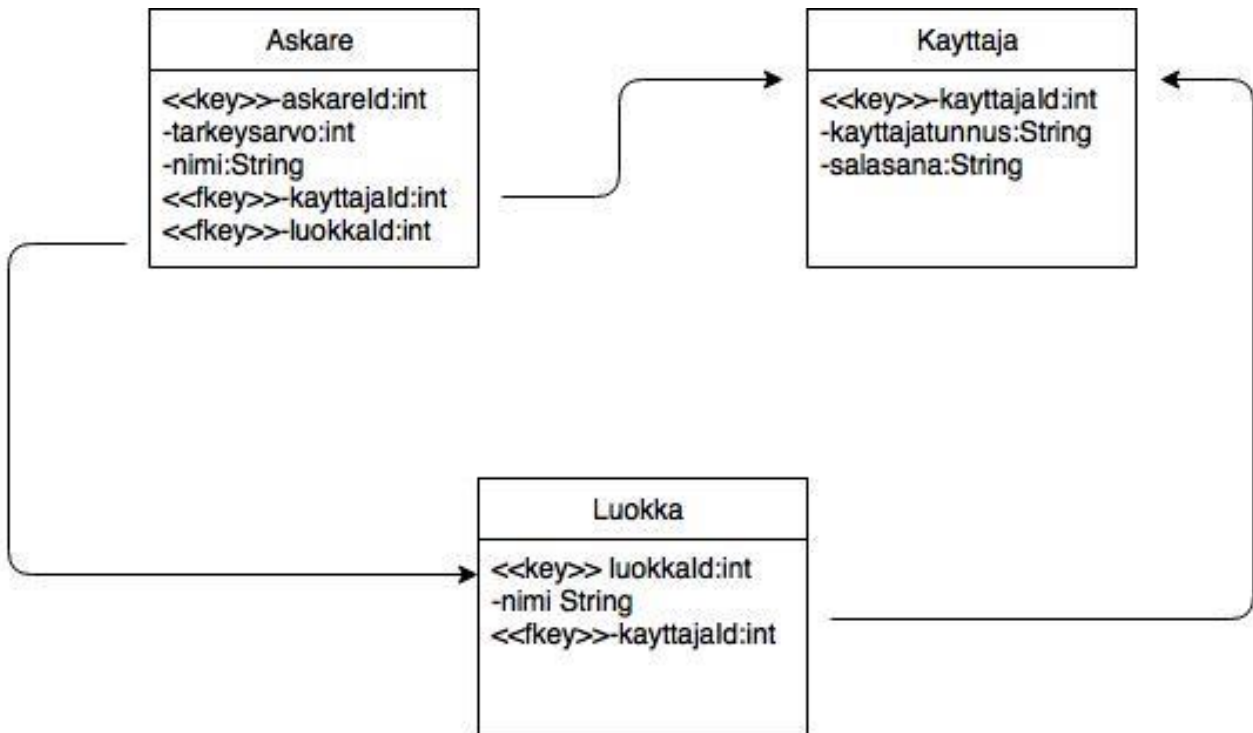
Kayttaja taulussa on jokaisen yksittäisen käyttäjän kirjautumistiedot ja uniikki id. Käyttäjällä voi olla lukuisia askareita ja luokkia.

Tietokohde: Luokka

Atribuutti	Arvojoukko	Selitys
luokkaid	SERIAL	Automaattisesti generoitu tunnus
nimi	varchar(40)	Käyttäjän asettama nimi luokalle
kayttajaid	integer	fkey kayttajaan

Luokka taulussa on luokan nimi, sekä viittaus kayttaja tauluun ja oma uniikki id jokaiselle yksittäiselle luokalle. Yhteen luokkaan voi kuulua useita askareita. Jokaisella luokalla on vain yksi käyttäjä.

4. Relaatietietokantakaavio

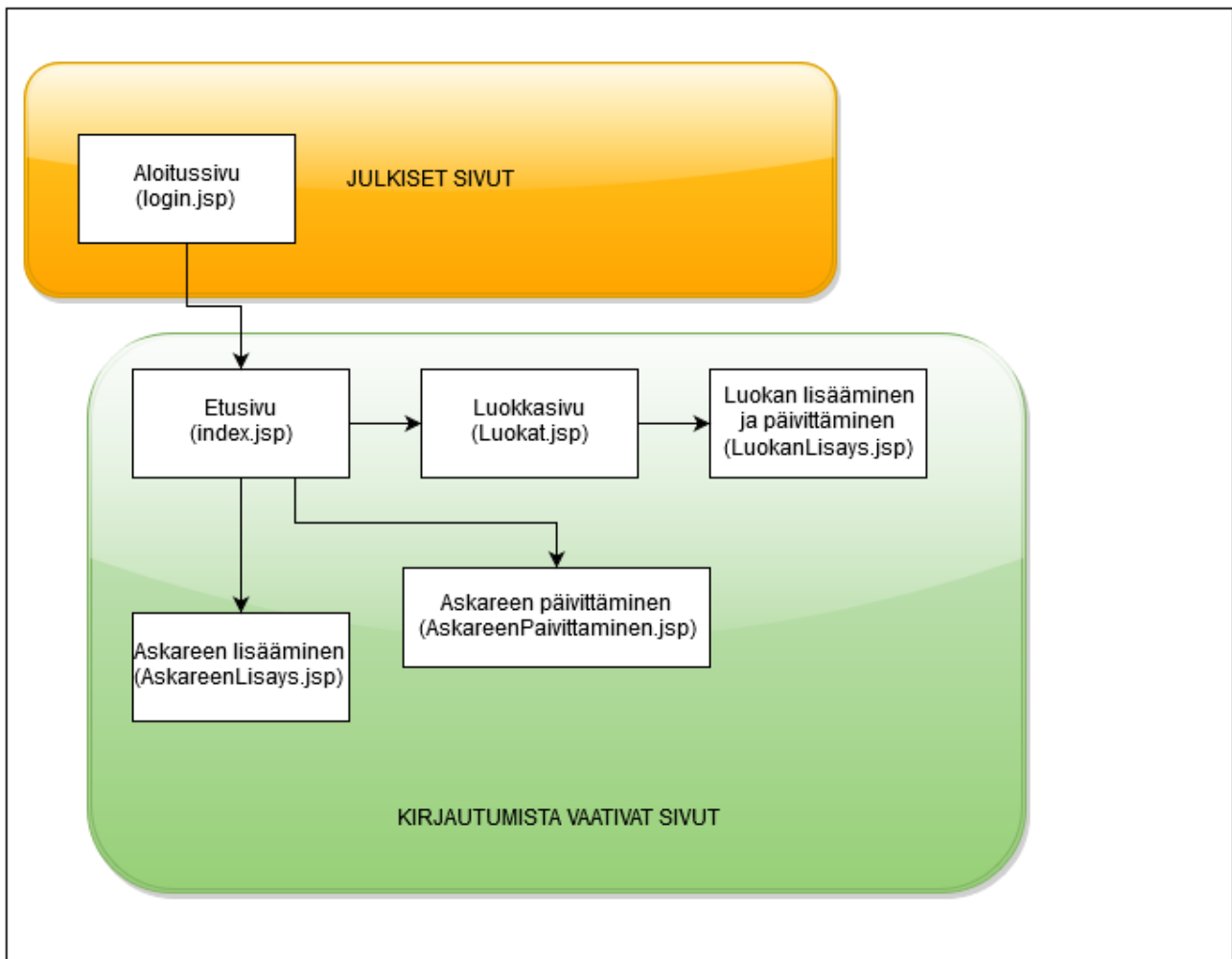


5. Järjestelmän yleisrakenne

Sovelluksen rakenne noudattaa MVC-arkkitehtuuria. Mallit luovat yhteyden tietokantaan ja sitä hyödyntämällä kontrollereissa pystytään tallettamaan tietokantaan tietoja näkymiltä saatujen tietojen mukaisesti. Näkymät taas näyttävät käyttäjälle sivun controllerilta saatujen tietojen mukaisesti.

Sovelluksessa näkymät ovat jsp tiedostoja, jotka sijaitsevat web kansion alla. Näkymissä hyödynnetään css-tyylisivuja ja perusnäkö on luotu pohja.tag tiedostoon, joka sijaitsee tags kansiossa. Mallit vuorostaan ovat Java luokkia jotka sijaitsevat Models kansion syövereissä ja controllerit ovat taas Javan servlettejä, jotka voi löytää Servlets kansioista.

6. Käyttöliittymä ja järjestelmän komponentit



7. Asennusohje

Sovelluksessa käytetään PostgreSQL tietokantaa, jota varten tarvitsee luoda tunnukset. Ohjeet tätä varten löytyvät osoitteesta www.postgresql.org. Sovelluksen oma taulurakenne löytyy sql kansion alla olevasta create-tables.sql tiedostosta, jonka voi löytää GitHubista osoitteesta <https://github.com/GandhiCorn/Muistilista>.

Tomcat palvelimelle asennus tapahtuu viemällä dist kansion Muistilista.war tiedosto webapps hakemistoon.

Sovellus löytyy valmiiksi asennettuna osoitteesta t-tuharo@users.cs.helsinki.fi/Muistilista

8. Käynnistys- / käyttöohje

Käynnistä sovellus osoitteesta t-tuharo@users.cs.helsinki.fi/Muistilista/Kirjautuminen

valmiit tunnukset ja salasanat:

kayttaja, salasana

kayttaja1, salasana1

Käyttöliittymä

ETUSIVU / REKISTERÖITYMINEN

↳ sisään kirjautuminen

↳ ulos kirjautuminen

↳ MUISTILISTA SIVU

↳ lisää askare

↳ poista askare

↳ muokkaa askareen tietoa

↳ tallennus

↳ LUOKAT (lista)

↳ lisää luokka

↳ muuta luokkaa (lista luokan askareista)

↳ luokan poisto

↳ tallennus