

Lemmikkitietokanta

TIETOKANTASOVELLUS

Kim Martesuo

HELSINGIN YLIOPISTO | 2015

Sisällysluettelo

1. Johdanto
 - 1.2 Lemmikitietokanta
 - 1.3 Toteutus
2. Yleiskuva järjestelmästä
 - 2.1 Käyttötapaukset
7. Asennustiedot
8. Käynnistys- / käyttöohje
9. Testaus, tunnetut bugit, puutteet ja jatkokehitysideat
10. Omat kokemukset

1. Johdanto

1.2 Lemmikitietokanta

Lemmikitietokantaan rekisteröityneet käyttäjät voivat lisätä lemmikkejään, luonnin yhteydessä lemmikille luodaan oma sivu. Käyttäjät voivat selailla omia ja muiden käyttäjien lemmikkejä ja hakea lemmikkejä esimerkiksi rodun perusteella.

Käyttäjät voivat myös tarkistaa palvelusta käyttäjän lähellä asuvia lemmikkejä postinumeron perusteella. Käyttäjät voivat lisätä lemmikkinsä ryhmiin, jotka ovat luotu käyttäjien toimesta.

Keskeisimpiä toimintoja ovat:

- Kirjautuminen
- Lemmikin luominen/Muokkaus/Poisto
- Lemmikkien haku
- Lähellä asuvat lemmikit
- Ryhmiin lisääminen
- Ryhmän luominen

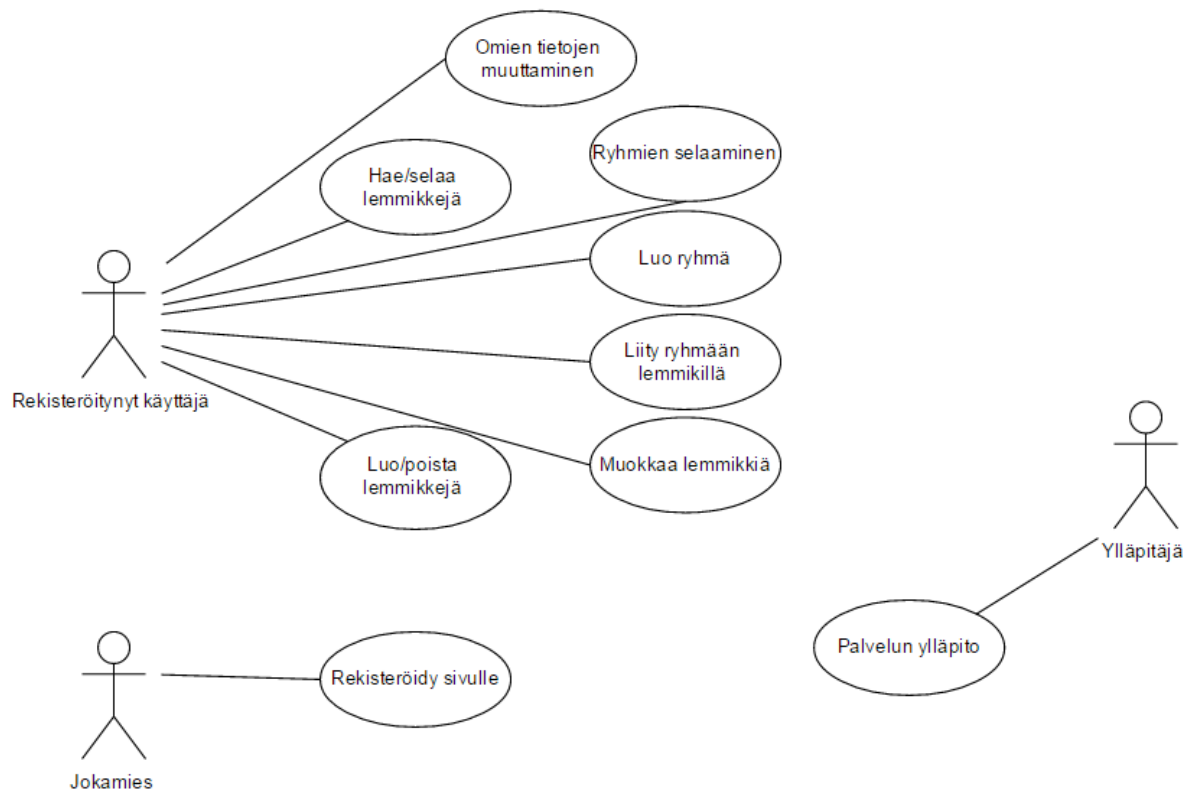
1.3 Toteutus

Sovelluksen toteuttamiseen käytän Java ja HTML ohjelmointikieliä. Tietokantana toimii PostgreSQL. Ympäristönä toimii laitoksen users-palvelin Apache/Tomcat palvelimen alla.

2. Yleiskuva järjestelmästä

2.1 Käyttötapaukset

Seuraava käyttötapauskaavio näyttää miten järjestelmän sidosryhmät liittyvät järjestelmään. Sidosryhmiä ei ole kovin montaa, sillä kaikille käyttäjille on samat palvelut tarjolla.



Käyttäjäryhmiä ovat:

Jokamies

On kuka tahansa vierailija sivulla. Hän voi rekisteröityä sivulle pystyäkseen käyttää kaikkia palveluita, tai vain käydä katsomassa sivua. Ylläpitäjä ja rekisteröitynyt käyttäjä kuuluvat myös tähän sidosryhmään.

Rekisteröitynyt käyttäjä

Voi käyttää kaikkia tarjolla olevia palveluita. Rekisteröitynyt käyttäjä luo siis itselle tilin palveluun.

Ylläpitäjä

Tällä sidosryhmällä tarkoitetaan sivua ja tietokantaa ylläpitäviä henkilöitä. Ylläpitäjä voi tietysti tehdä mitä vaan palvelulla.

Käyttötapaukset ovat:

Rekisteröityminen sivulle

Kuka tahansa voi rekisteröityä sivulle käyttääkseen palveluita.

Omien tietojen muuttaminen

Rekisteröitynyt käyttäjä voi luonnollisesti muuttaa tietojaan.

Hae/selaa lemmikkejä

Tällä käyttötapauksella tarkoitetaan, että käyttäjä voi hakukenttää käyttäen hakea lemmikkejä, mitkä täyttävät haun kriteerit. Haku suoritetaan tekemällä SQL-kysely tietokantatauluun, mikä sisältää kaikki lemmikit. Käyttäjä voi esimerkiksi hakea samalla alueella asuvia samaa rotua olevia lemmikkejä. Käyttäjä näkee lemmikin omistajan tiedot ja voi halutessaan vaikka sopia tapaamisen lähimpään koirapuistoon.

Luo/poista lemmikkejä

Käyttäjä voi luoda lemmikkejä. Tarkoituksena siis on, että luodaan oma tietokantainsertti omasta oikeasti olemassa olevasta lemmikistä. Lemmikin voi tietysti myös poistaa tietokannasta. Käyttäjällä voi olla monta lemmikkiä.

Muokkaa lemmikkiä

Lemmikin tiedot saattavat muuttua, joten niitä pystyy muokata.

Liity ryhmään

Käyttäjä voi liittää lemmikkinsä ryhmään. Esimerkiksi käyttäjä voi liittää kissansa ”Itä-Helsingin kissat” ryhmään.

Luo ryhmä

Käyttäjät voivat luoda ryhmiä, mihin muuta käyttäjät voivat liittyä.

Ryhmien selaamisen

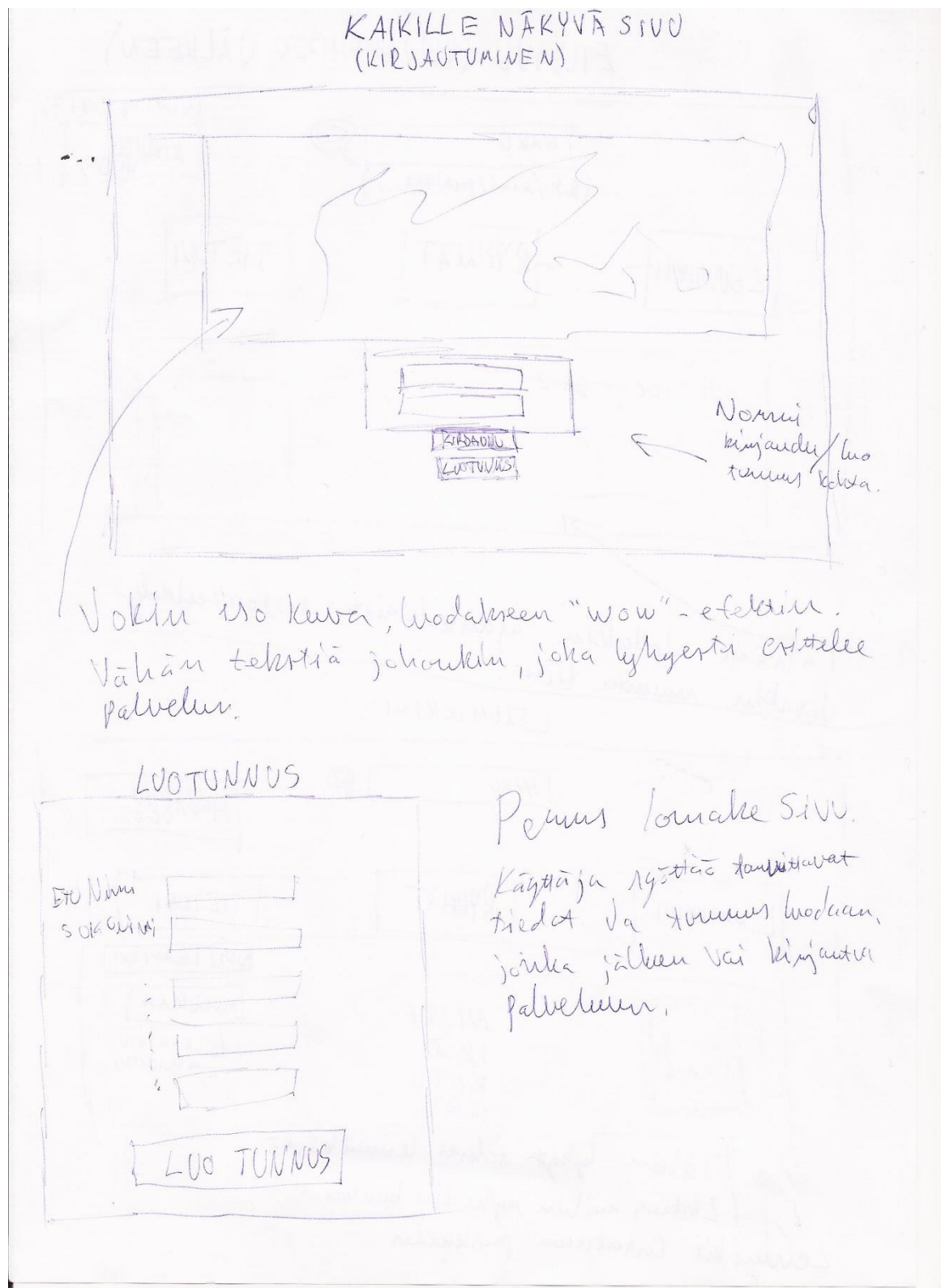
Käyttäjät voivat lemmikkien tavoin myös selata/hakea ryhmiä.

Palvelun ylläpito

Ylläpitäjä ylläpitää palvelua.

2.2 Käyttöliittymäkaaviot

Piirsin käyttöliittymäkaaviot paperille kynällä. Näin pystyin nopeasti luomaan pohjan sivujen ulkonäölle.



ETUSIVU (KIRJASTUMISEN JÄLKEEN)

HAKU
(Rotu/nimi/potkuvko...)

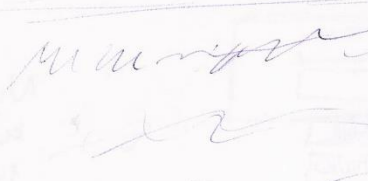
(NIMIMERKKI?)

KIRJASTU
 ULOS

LEMMIKKINI

RYHMÄT

TIETONI



Tähän voitakin ajankohtaista tietoa/esityksiä...
 Voitakin muuta tietoa.

LEMMIKKINI


HAKU

KIRJASTU
 ULOS

LEMMIKKINI

RYHMÄT

TIETONI



NIMI
 IKÄ
 ROTU
 VÄRI

UUSI LEMMIKKI
 MOOKKAA
L LEMMIKKIN
M UOKKAA

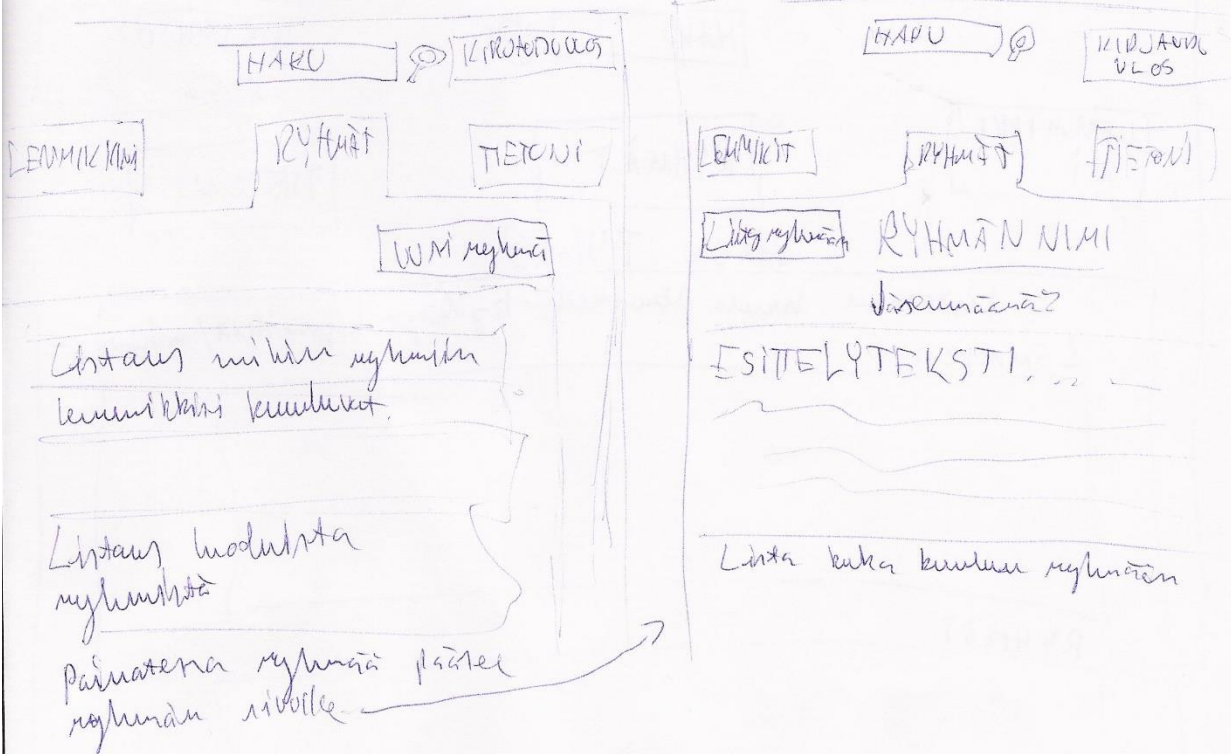
Tähän lyhyt teksti lemmikestä.
 Laitaus niihin ryhmäsiin kuuluu.
 Lemmikit listataan peräkkäin

Hand-drawn sketch of a database schema for a library system. The sketch shows the following tables and fields:

- HAKU**: A table with a single field.
- KIRJASTOVLGS**: A table with a single field.
- LEMMIKKINI**: A table with a single field.
- RYHMAT**: A table with a single field.
- TIETONI**: A table with a single field.
- KIRJASTOLEMMIKKI**: A table with a single field.
- TALOUS**: A table with a single field.
- NIMI, IKA, KOTU, VAP**: A group of fields, likely representing personal information.
- TEKSTI**: A table with a single field.

VO91 LERNMILCH
Sammelkassen NUV kein Lernmilch nachweis NUV.
DVL/ST

RHYMAT



UUSI RYHMÄ/LOMAKE

HAUO ☐ KIRJASTO ULOS

LEMMIKKI RYHMÄ TIETONI

Ryhmän nimi:

Esittely:

TALLENNUS
POISTA

Muita tietoja voi lisätä myöhemmin

TIETONI

HAUO ☐ KIRJASTO ULOS

LEMMIKKI RYHMÄ TIETONI

KÄYTTÄJÄTUNNUS

ETUNIMI

SOKUNIMI

POSTINUMERO

TALLENNUS

Hiljääksi tietoja ei
tunnista. Pelkkä postinumero
mittää, että löytyy läheltä
asuvat lemmikit.

HAKOSIVU

HAUO ☐ KIRJASTO ULOS

LEMMIKKI RYHMÄ TIETONI

HAUN TULOS

Listataan haun perusteella löydetty lemmikit/ryhmät

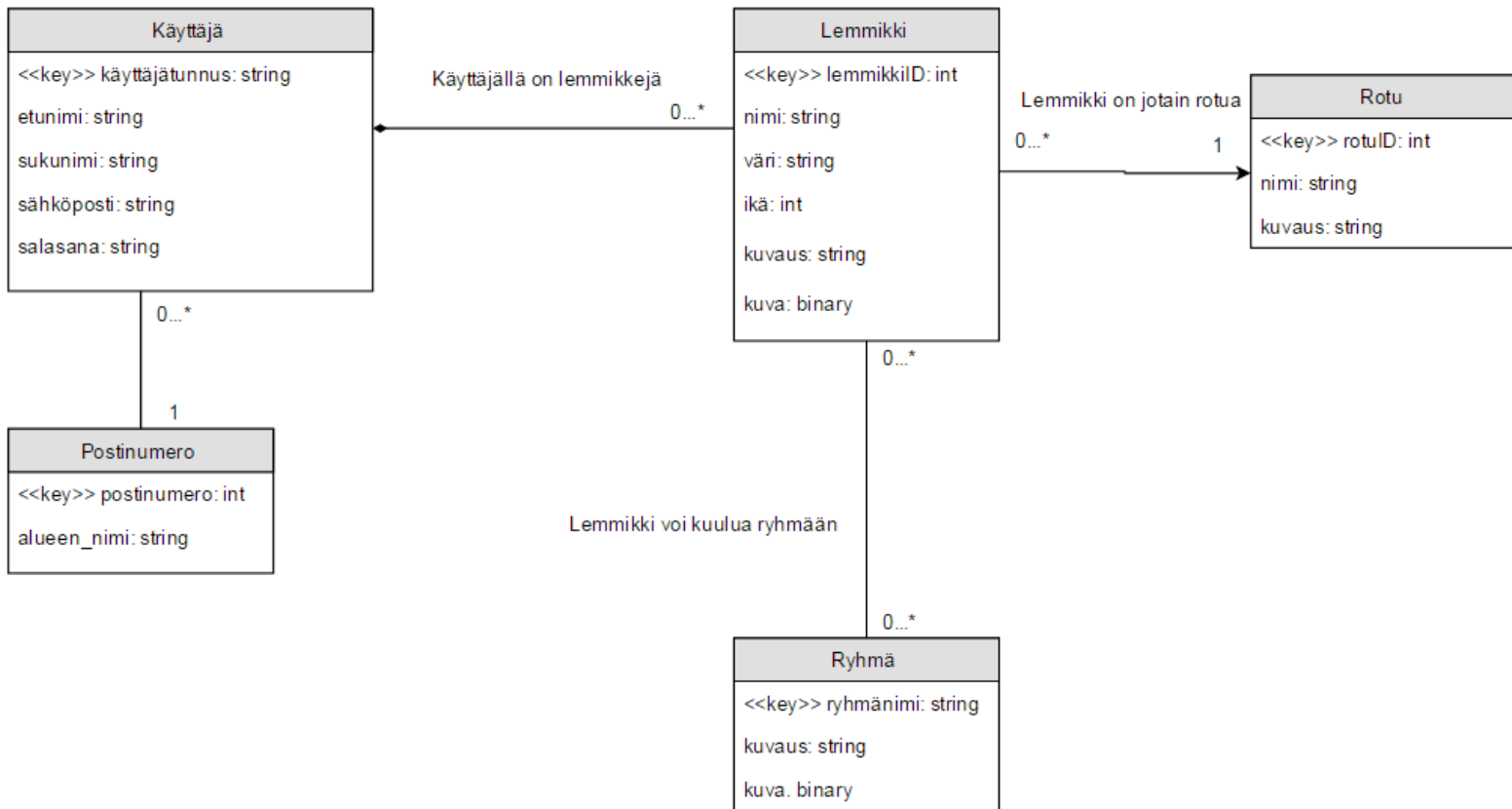
LEMMIKIT

RYHMÄT

3. Järjestelmän tietosisältö

3.1 Käsitekaavio

Tein käsitekaavion draw.io:lla. Tietokohteita on yhteensä viisi. Myöhemmin tulee ehkä joitain pieniä muokkauksia tarpeen vaatiessa.



Käyttäjä

Käyttäjä tietokohde koostuu rekisteröityneen käyttäjän tiedoista. Käyttäjän luoma käyttäjätunnus toimii pääavaimena, joten samoja käyttäjätunnuksia ei voi olla. Käyttäjällä voi olla monta lemmikkiä. Käyttäjällä on vain yksi postinumero, jota käytetään lähellä asuvien lemmikkien hakemiseen.

Postinumero

Tämä tietokohde koostuu vain postinumerosta, joka toimii pääavaimena, ja alueen nimi. Postinumerolla voi olla monta käyttäjää.

Lemmikki

Luonnin yhteydessä lemmikille annetaan lemmikki-ID joka toimii pääavaimena. Muita lemmikin attribuutteja ovat nimi, väri ja ikä. Lemmikki voi olla yhtä rotua. Käyttäjä voi myös lisätä kuvan lemmikistä ja kuvauksen lemmikistä.

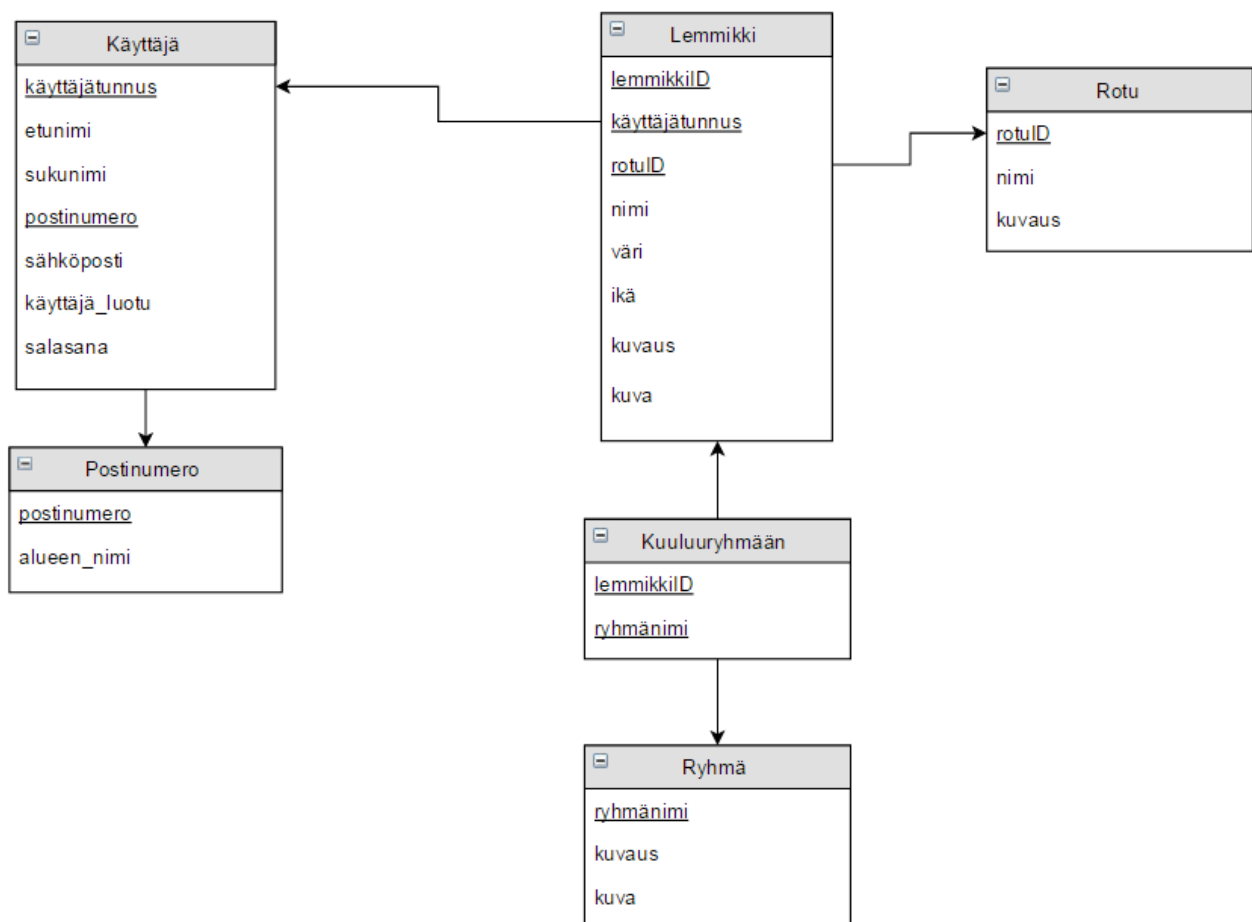
Rotu

Pääavaimena toimii rotu-ID. Muita attribuutteja ovat nimi ja kuvaus. Rodulla voi olla monta lemmikkiä.

Ryhmä

Ryhmän pääavain on nimi, joten samannimisiä ryhmiä ei voi olla. Ryhmällä on kuvaus ja ehkä kuva.

4. Relaatiotietokantakaavio



5. Järjestelmän yleisrakenne

Tietokantasovellus noudattaa MVC-mallia. Mallin eri osat on sijoitettu seuraavasti:

Mallit: hakemisto lemmikkitietokanta.Models. Java-tiedostona

Kontrollerit: hakemisto lemmikkitietokanta.Servlets. Java-tiedostona

Näkymät: Web Pages. Jsp-tiedostoina

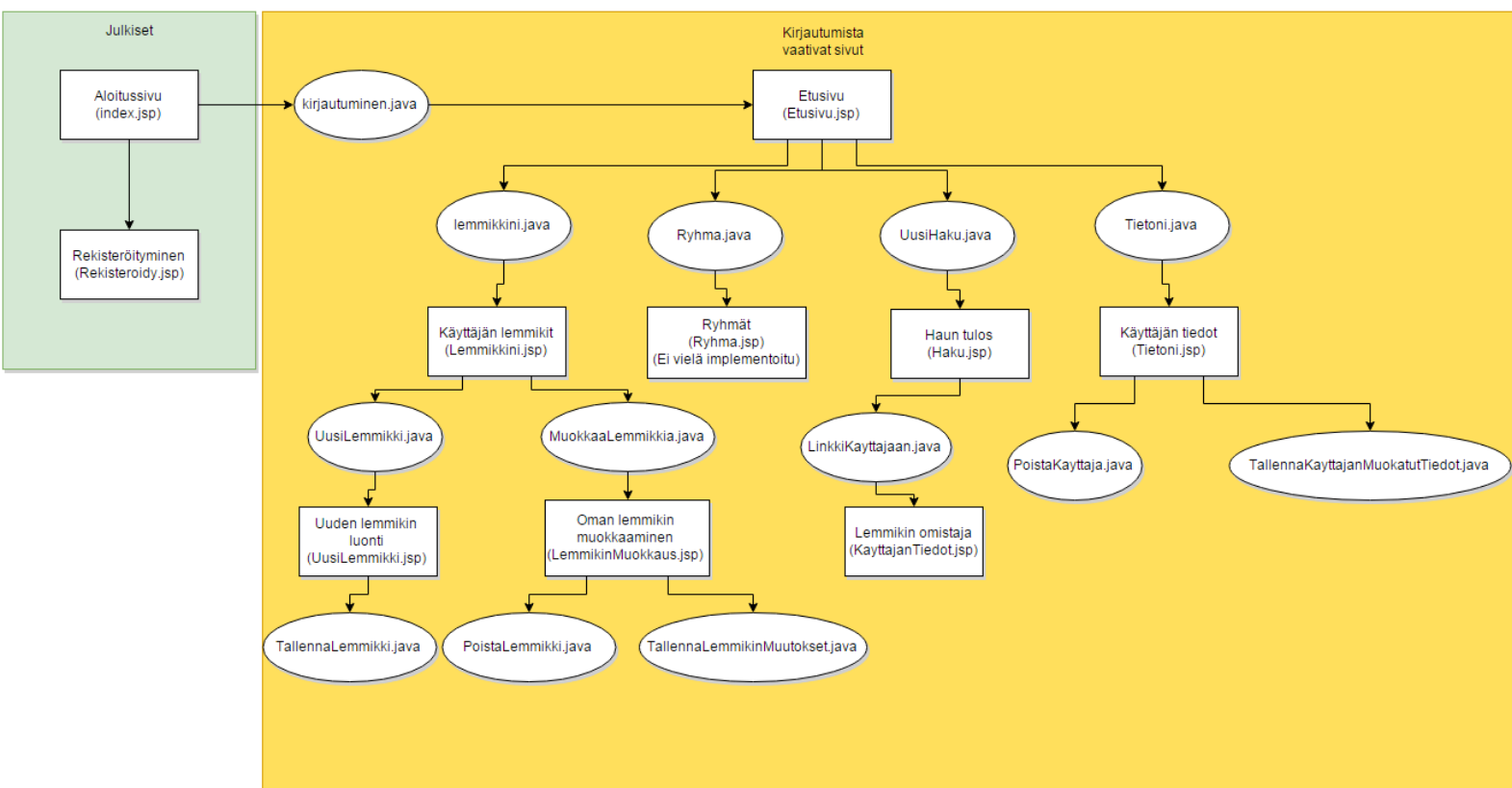
Css-kansio sisältää css-tiedostot. META-INF sisältää tiedoston context.xml, mikä sisältää tietokannan tiedot. WEB-INF sisältää tiedoston web.xml, mikä sisältää tiedot mihin servletit johtavat. WEB-INF kansio sisältää myös tags-kansion, johon on sijoitettu näkymien pohjatiedostot, eli tag-tiedostot. lemmikkitietokanta.Models sisältää myös muutaman staattisen metodin naytaJSP-luokassa ja tietokantaan yhteyden ottamista varten luokan tietokantayhteys.

Kirjautumisen aikana luodaan istunto johon sisälletään kirjautuneen käyttäjätunnus. Tämän avulla voidaan sitten hakea tietokannasta tarvittavat tiedot, kun tehdään listauksia, koska käyttäjätunnus on pääavain käyttäjälle.

6. Käyttöliittymä ja järjestelmän komponentit

Alla olevasta kaaviosta suorakulmiot ovat käyttäjälle näkyviä sivuja. Pallot esittävät servlettiä, jonka kautta kuljetaan.

Kirjautumista vaativissa sivuissa on navigaatiopalkki, sitä ei oteta tässä huomioon. Tämä tarkoittaa sitä, että jokaisesta jsp-sivusta pääsee käyttäjän lemmikki-, ryhmä- ja käyttäjän tietosivulle.



7. Asennustiedot

Tällä hetkellä sovellus on asennettu Helsingin Yliopiston users-palvelimelle tomcatin avulla. sovellus on Lemmikitietokanta.war muodossa ja se on sijoitettu tomcat/webapps-kansioon.

Mikäli haluaa asentaa sovelluksen muualle täytyy tehdä tarvittavat muutoksen Lemmikitietokanta.xml tiedostoon, jotta saa oman tietokannan toimimaan sovelluksella. Tiedoston sijainti on tomcat/conf/catalina/localhost/Lemmikitietokanta.xml. Itse sovellus tulee sijoittaa tomcat/webapp kansioon, mikäli käyttää apunaan tomcattia.

8. Käynnistys- / käyttöohje

Sovellukseen pääsee menemällä nettiselaimella osoitteeseen:

t-kmartesu.users.cs.helsinki.fi/Lemmikitietokanta

Sisään voi kirjautua testitunnuksilla käyttäjä: user salasana: password.

9. Testaus, tunnetut bugit, puutteet ja jatkokehitysideat

Sovellus ei ole järin suuri, joten syvällisempää testausta ei ole tarvittu. Olen testannut sovellusta itse ja olen myös pyytänyt muutamaa ystävää testaamaan sovellusta, he löysivät muutaman virheen mitkä sitten korjasin.

Sovelluksesta puuttuu vielä ryhmiin liittymisen ominaisuus. Tämän toteuttaminen pitäisi onnistua suhteellisen helposti. Sovelluksen käyttöliittymää voisi parannella, mutta käyttöliittymä ei ollut tämän kurssin keskiössä.

10. Omat kokemukset

Pidin tästä kurssista paljon, koska opin paljon uutta. Olin kiinnostunut siitä, miten tällaisia sovelluksia ylipäättään toteutetaan ja tämän kurssin avulla sain siitä selvän kuvan. Olisi ollut hankalampaa lähteä itse opiskelemaan aihetta ilman ohjausta.

Vaikein asia kurssilla oli minun kohdaltani käyttöliittymän toteuttaminen. En ollut aikaisemmin luonut nettisivuja, joten sivujen toteuttamiseen meni jonkun verran aikaa, mikä myös näkyy sivujen ulkonäössä.

Huomasin myös miten suunnitelma ja itse sovelluksen toteuttaminen ei mennyt yksi yhteen. Olin suunnitellut sovellukselle enemmän ominaisuuksia kuin mitä ehdin toteuttamaan. Tämä johtui ehkä siitä, että en ollut aikaisemmin tehnyt tietokantasovelluksia.

Vaivalloista oli myös se, että tietokantaan ei saanut yhteyttä NetBeansin kautta, jos työskenteli omalla koneella kotona. Uusimman version joutui aina siirtämään palvelimelle, jotta sai testattua sovellusta.

Tahdon kiittää ohjeista, jotka olivat erittäin hyvät ja niiden myötä kotona työskentely onnistui hyvin.

Kiitos hyvästä kurssista!