

# **Kuvagalleria - Suunnitteludokumentti**

Sami Saada <saada@cs.helsinki.fi>

Helsinki 29.4.2009

Tietokantasovellus

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Tiedekunta — Fakultet — Faculty		Laitos — Institution — Department	
Matemaattis-luonnontieteellinen		Tietojenkäsittelytieteen laitos	
Tekijä — Författare — Author			
Sami Saada <saada@cs.helsinki.fi>			
Työn nimi — Arbetets titel — Title			
Kuvagalleria - Suunnitteludokumentti			
Oppiaine — Läroämne — Subject			
Tietojenkäsittelytiede			
Työn laji — Arbetets art — Level		Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Tietokantasovellus		29.4.2009	16 sivua
Tiivistelmä — Referat — Abstract			
<p>Tietokantasovelluksen suunnitteludokumentti. Dokumentissa kerrotaan tietokantasovelluksen rakenteesta ja miten se tulisi toteuttaa. Dokumentti sisältää erilaisia oleellisia kaavioita.</p>			
Avainsanat — Nyckelord — Keywords			
Tietokantasovellus, Suunnitteludokumentti			
Säilytyspaikka — Förvaringsställe — Where deposited			
Muita tietoja — övriga uppgifter — Additional information			
Tämä on tietokantasovelluksen suunnitteludokumentti			

# Sisältö

<b>1</b>	<b>Johdanto</b>	<b>1</b>
1.1	Järjestelmän tarkoitus . . . . .	1
1.2	Toimintaympäristö . . . . .	1
1.3	Rajaukset . . . . .	1
1.4	Toteutusympäristö . . . . .	2
<b>2</b>	<b>Yleiskuva järjestelmästä</b>	<b>3</b>
2.1	Sidosryhmäkaavio . . . . .	3
2.2	Käyttäjryhmät . . . . .	3
2.2.1	Vierailija . . . . .	3
2.2.2	Rekisteröitynyt käyttäjä . . . . .	3
2.2.3	Ylläpito . . . . .	3
<b>3</b>	<b>Käyttötapaukset</b>	<b>4</b>
3.1	Vierailijan käyttötapaukset . . . . .	4
3.1.1	Kuvien selaaminen . . . . .	4
3.1.2	Kuvien hakeminen . . . . .	4
3.1.3	Kuvien tietojen katsominen . . . . .	5
3.1.4	Apusivun lukeminen . . . . .	5
3.2	Rekisteröityneen käyttäjän käyttötapaukset . . . . .	5
3.2.1	Rajattujen kuvien katsominen . . . . .	5
3.2.2	Profiili . . . . .	5
3.3	Ylläpidon käyttötapaukset . . . . .	5
3.3.1	Käyttäjähallinta . . . . .	5
3.3.2	Kuvien hallinta . . . . .	6
3.3.3	Kategorioiden hallinta . . . . .	7
<b>4</b>	<b>Järjestelmän tietosisältö</b>	<b>8</b>
4.1	Käsitekaavio . . . . .	8
4.2	Tietosisällön kuvaus . . . . .	8
<b>5</b>	<b>Käyttöliittymän hahmotelma</b>	<b>10</b>
5.1	Hahmotelma . . . . .	10

5.2	Näkymien väliset yhteydet . . . . .	10
<b>6</b>	<b>Relaattiotietokanta</b>	<b>11</b>
6.1	Relaattiotietokantakaavio . . . . .	11
6.2	SQL-lauseet . . . . .	11
6.2.1	Create Table . . . . .	11
6.2.2	Create View . . . . .	14

# 1 Johdanto

Järjestelmä on kuvagalleria ja tarkoitus on tutustuttaa sen saloihin pikaisella pintaraapaisulla.

## 1.1 Järjestelmän tarkoitus

Järjestelmä tulee toimimaan kuvagalleriana, jossa voi selata kuvia eri tasoilla oikeuksilla. Kuvia on tietenkin mahdollista lisätä ja niiden tietoja pystyy muuttamaan.

Järjestelmä on tarkoitettu niin yksityishenkilöiden kuin yritysten ja yhdistysten kuvallisen toiminnan tukemiseksi. Erilaisten järjestettyjen tapahtumien kuvat saadaan helposti esille. Käyttäjätasojen ansiosta myös rajatun piirin keskuuteen tarkoitetut kuvat onnistuvat.

Järjestelmän päätavoite on ratkaista sitä käyttävien tahojen kuvien hallintaongelmat tai edes helpottaa niitä.

## 1.2 Toimintaympäristö

Järjestelmä toteutetaan käyttäen ohjelmointikielenä PHP5.2.x:tä ja tietokantana PostgreSQL:ää, joten se toimii missä tahansa ympäristössä, joka tukee tarpeeksi tuoretta PHP-versiota, jossa on käytetyt ominaisuudet. Ohjelmointikieli ei aseta rajoitteita käyttöjärjestelmälle.

Tietokantapalvelimelta vaaditaan jokin PHP:n PDO-moduulin ajurin tukema tietokantaluista. Tietokantapalvelin voi olla yhdistettynä sovelluspalvelimeen tai siitä erikseen. Molemmat vaihtoehdot ovat tuettuna sovelluksen toteutuksessa.

Tietokannan luontikyselyt kirjoitetaan PostgreSQL:ää ajatellen, mutta ne on mahdollista saada pienehköllä vaivalla tukemaan myös muita tietokantaluustoja. Kaikki nykyiset käyttöjärjestelmät tukevat tietokantaluustoja kohtuullisen laajalti.

Järjestelmä toteutetaan esittelykuntoon TKTL:n tietokantasovelluspalvelin Alkokrunille (*db.cs.helsinki.fi*).

Asiakaspuolelta vaaditaan tuki XHTML ja CSS-standardeille sekä tuki JavaScriptille. Työhön kuuluu osia WEB 2.0:n tekniseltä puolelta.

## 1.3 Rajaukset

Sovellus toteutetaan sisältämään kaikki siihen määritellyt ominaisuudet. Se on mahdollista toteuttaa kokonaan annetussa ajassa, joten koko määrittelyn toteuttaminen ei ole ongelma.

Esittelytilaisuudessa on melko rajattu määrä tietoa, sillä tekijänoikeuksia ei rikota.

Tietoa on sen verran, että kaikkia ominaisuuksia on mahdollista esitellä.

Asiakkaan selaimen on tuettava toimintaympäristön määrittelemiä tekniikoita, joten Internet Explorer 6 -selaimen tukeminen ei ole pääasiassa tässä projektissa.

Käytettävyyden parantamiseksi saatetaan käyttää CSS-standardin ulkopuolisia osia, joita nykyaikaiset selaimet kuitenkin tukevat.

## **1.4 Toteutusympäristö**

Työ toteutetaan omassa kehitysympäristössä, johon kuuluu Gentoo Linux, Apache 2, PHP 5.2.8-r2, PostgreSQL 8.2.7 ja vim-tekstieditori sopivilla lisäosilla.

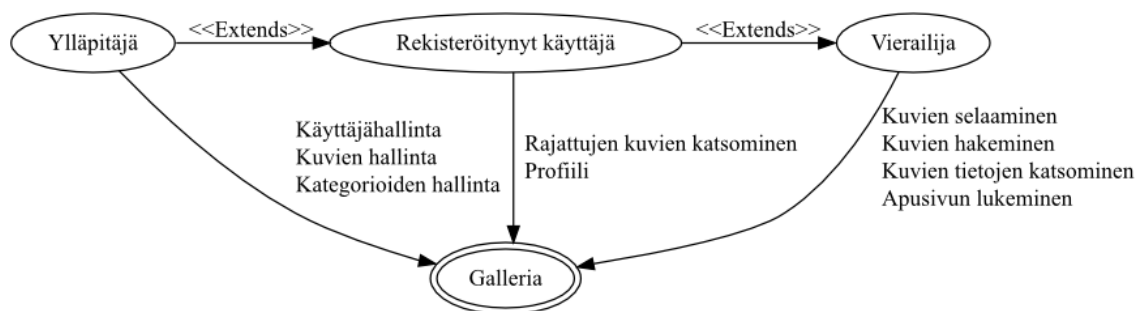
Käytössä on myös SVN-palvelin, jolle varastoidaan projektin eri vaiheet ja vain toimivat osat ovat näkyvissä toteutuspalvelimella.

Toteutusympäristön testausympäristö on pääasiallisesti Firefox 3 -selain, jossa on tärkeinä lisäosina Firebug sekä Web Developer.

## 2 Yleiskuva järjestelmästä

Järjestelmän yleiskuvaa kuvataan sidosryhmäkaavion ja käyttäjäryhmien avulla.

### 2.1 Sidosryhmäkaavio



Kuva 1: Sidosryhmäkaavio

### 2.2 Käyttäjäryhmät

Määritellään sidosryhmäkaaviossa esiintyvät käyttäjäryhmät.

#### 2.2.1 Vierailija

Vierailija on käyttäjä, joka ei ole kirjautunut sisälle.

#### 2.2.2 Rekisteröitynyt käyttäjä

Rekisteröitynyt käyttäjä on sisäänkirjautunut käyttäjä.

#### 2.2.3 Ylläpito

Ylläpito koostuu rekisteröityneistä käyttäjistä, jotka kuuluvat johonkin ryhmään, jolla on ylläpito-oikeus.

## 3 Käyttötapaukset

Järjestelmässä on useita eri käyttötapauksia.

### 3.1 Vierailijan käyttötapaukset

Vierailijalla on järjestelmän julkiseen käyttöön liittyvät käyttötapaukset.

#### 3.1.1 Kuvien selaaminen

Kuvia voi selata muutamalla eri tavalla, kuten tuoreimpien kuvien kautta, liitesanojen (tagien) kautta, kategorioiden kautta tai hakemalla.

Tuoreimmissa kuvissa on listattuna kymmenen tuoreinta kuvaa. Kuvien tuoreus määräytyy kuvan lisäämisen ajankohdan perusteella. Tuoreimpien kuvien listasta pääsee kuvaa painamalla sen tietoihin. Tuoreimmat kuvat ovat järjestelmän etusivulla.

Liitesanat, tuttavallisemmin tagit, ovat kuviin liitettyjä sanoja. Kuviin voi liittää useita tageja. Ne näkyvät etusivulla ja niiden koko määräytyy niiden yleisyyden perusteella. Tagista saa näkyville sen sisältämät kuvat, joista taas pääsee kuvien tietoihin.

Kuvia voi selata kategorioiden kautta. Ne voivat sisältää myös alakategorioita. Kategoriat ovat yleisluontoisia kuvauksia kuvista, esimerkiksi jokin rakennus, tapahtuma yms. Kategoriasta saa esiin siihen liittyvät kuvat, muttei sen alakategorioiden kuvia. Kuvalistasta pääsee kuvien tietoihin.

Kuvien selaamisesta hakemalla on tietoa käyttötapausten kohdassa **Kuvien hakeminen**.

#### 3.1.2 Kuvien hakeminen

Kuvien hakemisen voi tehdä laaja-alaisesti tai suppeasti. Haussa on kolme kenttää: teksti, kuvaaja ja kuvausvuosi.

Teksti-kenttä sisältää vapaan tekstihaun, joka hakee kuvan kuvauksesta sekä tageista.

Kuvaaja-kenttä sisältää kuvaajan tiedot. Tällä kentällä voi hakea yhtä tai useampaa kuvaajaa. Kentän täyttö kertoo, että valitaan vain siihen täsmäävien kuvaajien kuvat.

Kuvausvuoden valinta tuo kuvat kyseisiltä vuosilta. Karsinta toimii samalla tavalla kuin edellisissä kentissä.

Hakutulokset listautuvat, kuten muutkin listaukset ja niistä pääsee kuvien tietoihin. Tämä tehdään mahdollisimman samanlaiseksi kuin muilla sivuilla.



### 3.1.3 Kuvien tietojen katsominen

Vierailijat voivat katsoa julkisia kuvia, joita he ovat selanneet tai hakeneet.

Kuvista näkee tietoina itse kuvan, kategorian, valokuvaajan, kuvausvuoden kuvaavan tekstin ja tagit.

### 3.1.4 Apusivun lukeminen

Vierailija voi lukea apusivulta tietoa häntä kiinnostavista asioista. Apusivulla on esitelty eri käyttötapauksien mahdolliset keskeiset ongelmat.

## 3.2 Rekisteröityneen käyttäjän käyttötapaukset

Rekisteröityneellä käyttäjällä on samat käyttötapaukset kuin vierailijalla, mutta käyttöoikeudet vaikuttavat siihen, että rekisteröitynyt käyttäjä näkee enemmän kuvia, sillä hän voi kuulua johonkin ryhmään, johon on liitetty kuvia.

### 3.2.1 Rajattujen kuvien katsominen

Rekisteröitynyt käyttäjä näkee ryhmilleen asetetut kuvat, joita muut käyttäjät eivät näe.

Rajatut kuva näkyvät niille käyttäjille, joilla on sama ryhmä kuin kuvien kategorioilla. Tätä varten tarvitaan suhde kategorioiden ja käyttäjäryhmien välille.

### 3.2.2 Profili

Rekisteröitynyt käyttäjä voi muokata omia tietojaan, kuten nimeä ja salasanaa.

## 3.3 Ylläpidon käyttötapaukset

Ylläpidolla on samat käyttötapaukset kuin rekisteröityneellä käyttäjällä, mutta hän näkee kaikkien ryhmien kuvat.

### 3.3.1 Käyttäjähallinta

Ylläpito voi lisätä ja poistaa käyttäjiä ja käyttäjäryhmiä. Se voi liittää käyttäjiä ryhmiin ja muokata käyttäjien tietoja.

#### Käyttäjien lisääminen

Ylläpito voi lisätä uusia käyttäjiä järjestelmään. Se täyttää tarvittavat tiedot, eli tunnuksen, nimen ja salasanan. Tämän jälkeen tunnus on luotu ja valmis käytettäväksi. Järjestelmä ei kuitenkaan hyväksy kahta samanlaista tunnusta.

### **Käyttäjien muokkaaminen**

Ylläpito voi muokata käyttäjiä. Muokattavia asioita ovat tunnus, nimi, salasana ja aktiivisuus. Järjestelmä ei kuitenkaan hyväksy kahta samanlaista tunnusta. Aktiivisuus kertoo sen, onko tunnuksella oikeus tehdä sitä vastaavia tehtäviä vai onko se ns. jäähyllä.

### **Käyttäjien poistaminen**

Ylläpito voi poistaa käyttäjiä. Tällöin tunnus häviää ja kuvien viittaukset vaihtuvat.

### **Käyttäjryhmien lisääminen**

Ylläpito voi lisätä uusia käyttäjäryhmiä. Ryhmälle annetaan nimi ja sen voi asettaa ylläpitoryhmäksi.

### **Käyttäjryhmien muokkaaminen**

Ylläpito voi muuttaa ryhmän nimeä ja sen ylläpito-oikeutta.

### **Käyttäjryhmien poistaminen**

Ylläpito voi poistaa ryhmän, jolloin kaikki siihen viittaavat käyttäjät poistetaan kyseisestä ryhmästä.

## **3.3.2 Kuvien hallinta**

Ylläpito voi lisätä, muokata ja poistaa kuvia, asettaa kuviin kategorioita ja tageja (yksi kategoria ja monta tagia). Jokainen muutos tallennetaan kuvan muokkaushistoriaan.

### **Kuvien lisääminen**

Ylläpito lisää kuvia järjestelmään täyttämällä kuvaan kategorian, kuvaajan, kuvausvuoden, sanallisen kuvauksen ja tageja.

### **Kuvien muokkaaminen**

Ylläpito voi muuttaa kuvien kuvausta, kategoriaa, kuvaajaa, kuvausvuotta ja siihen liittyviä tageja. Itse kuvadataa ei voi muuttaa huonotyyllisen käytön estämiseksi.

Kuvien näkyvyyttä voi myös muokata, jolloin voidaan päättää onko kuva näkyvissä vai ei.

### **Kuvien poisto**

Ylläpito voi poistaa kuvia galleriasta. Tällöin kaikki kuvaan liittyvät tiedot poistetaan.

### **Muokkaushistoria**

Muokkaushistoria päivittyy aina, kun kuva lisätään tai siihen tehdään jotain muutoksia. Muokkaushistoria näytetään kuvakohtaisesti ylläpidolle.

### **3.3.3 Kategorioiden hallinta**

Ylläpito voi lisätä ja poistaa kategorioita sekä asettaa käyttäjäryhmiä kategorioihin.

#### **Kategorioiden lisäys**

Ylläpito voi lisätä uusia kategorioita, joihin tämän jälkeen voi lisätä kuvia ja käyttäjäryhmiä.

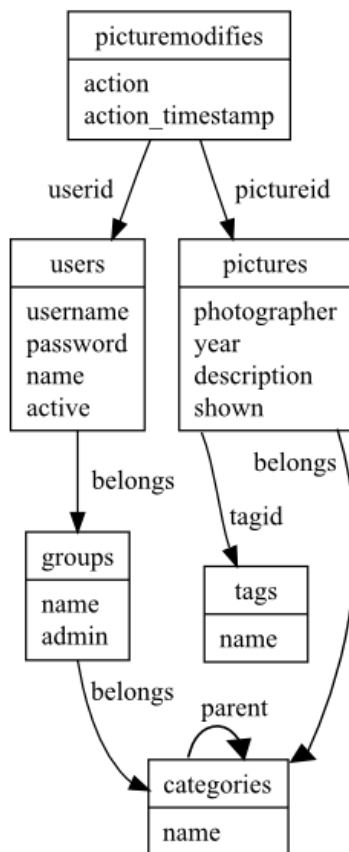
#### **Kategorioiden poisto**

Ylläpito voi poistaa kategorioita. Jos poistettava kategoria sisältää käyttäjäryhmiä, näiden välinen yhteys poistetaan. Jos poistettava kategoria sisältää kuvia, kysytään mihin liitetyt kuvat siirretään, jos tämä ei onnistu, niin kategoriaa ei anneta poistaa.

## 4 Järjestelmän tietosisältö

Kuvataan järjestelmän tietosisältö käsitekaavion kautta ja eritellään tarkemmin tietosisältöä.

### 4.1 Käsitekaavio



Kuva 2: Käsitekaavio

### 4.2 Tietosisällön kuvaus

Käsitellään käsitekaavion tietosisällöt.

Käyttäjät		
Attribuutti	Arvojoukko	Kuvailu
Nimi	merkkijono	Kertoo käyttäjän nimen
Käyttäjätunnus	merkkijono	Käyttäjän tunnus, jolla kirjautuu sisään
Salasana	hajautettu merkkijono	Käyttäjän salasana suojattuna
Aktiivisuus	totuusarvo	kertoo, onko käyttäjä aktiivinen

Käyttäjärhyhmät		
Attribuutti	Arvojoukko	Kuvailu
Nimi	merkkijono	Kertoo ryhmän nimen
Ylläpito	boolean	Kertoo, onko ryhmä osa ylläpitoa

Kategoriat		
Attribuutti	Arvojoukko	Kuvailu
Nimi	merkkijono	Kertoo kategorian nimen

Kuvat		
Attribuutti	Arvojoukko	Kuvailu
Kuvaaja	merkkijono	Kertoo, kuka on kuvannut kuvan
Kuvausvuosi	merkkijono	Kertoo, milloin kuva on otettu
Tekstikuvaus	merkkijono	Kuvaa kuvaa sanallisesti
Näkyvyysaste	totuusarvo	Kertoo, onko kuva näkyvilä

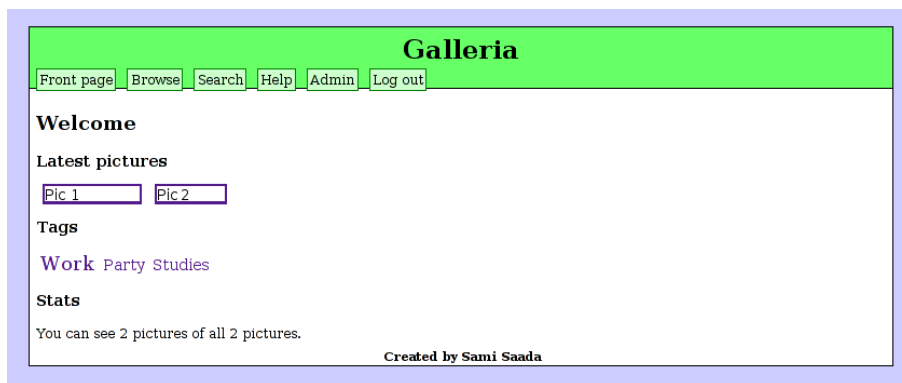
Kuvamuutokset		
Attribuutti	Arvojoukko	Kuvailu
Kuvaviite	kokonaisluku	Kertoo viitenumeron kuvaan
Käyttäjäviite	kokonaisluku	Kertoo viitenumeron käyttäjään
Toiminto	merkkijono	Kertoo millaisesta toiminnosta kyse
Toiminnon aikaleima	aikaleima	Kertoo milloin toiminto tehtiin

Tagit		
Attribuutti	Arvojoukko	Kuvailu
Nimi	merkkijono	Kertoo tagin nimen

## 5 Käyttöliittymän hahmotelma

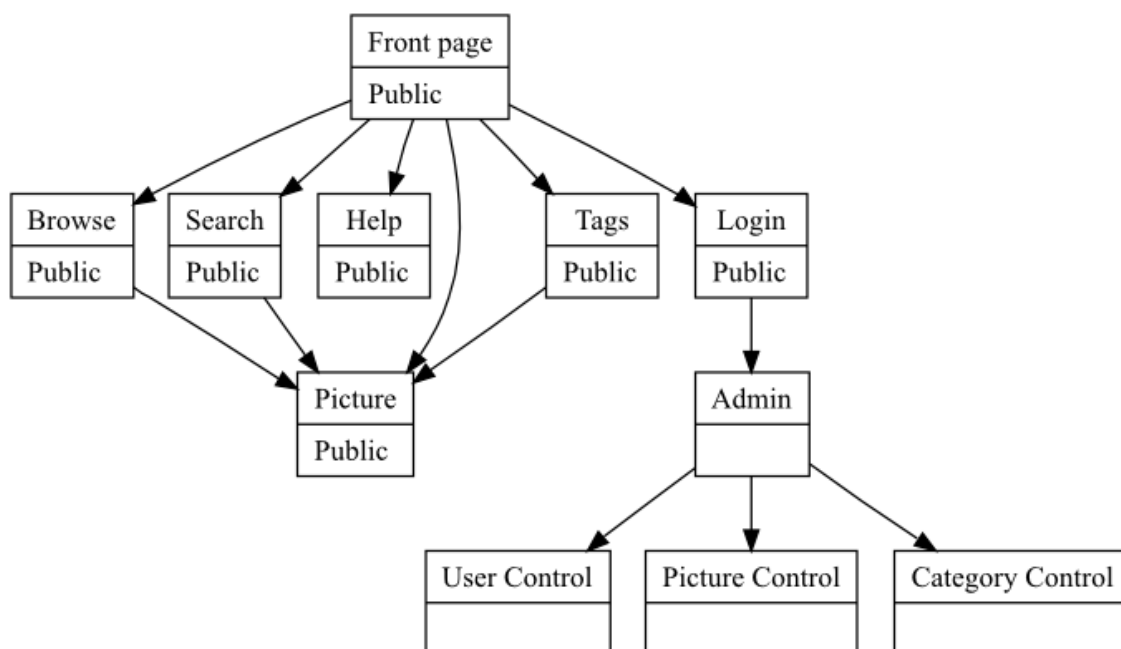
Käyttöliittymästä on hahmotelmakuva ja kaavio, jossa on kerrottu eri näkymien väliset yhteydet.

### 5.1 Hahmotelma



Kuva 3: Etusivun näkymä

### 5.2 Näkymien väliset yhteydet

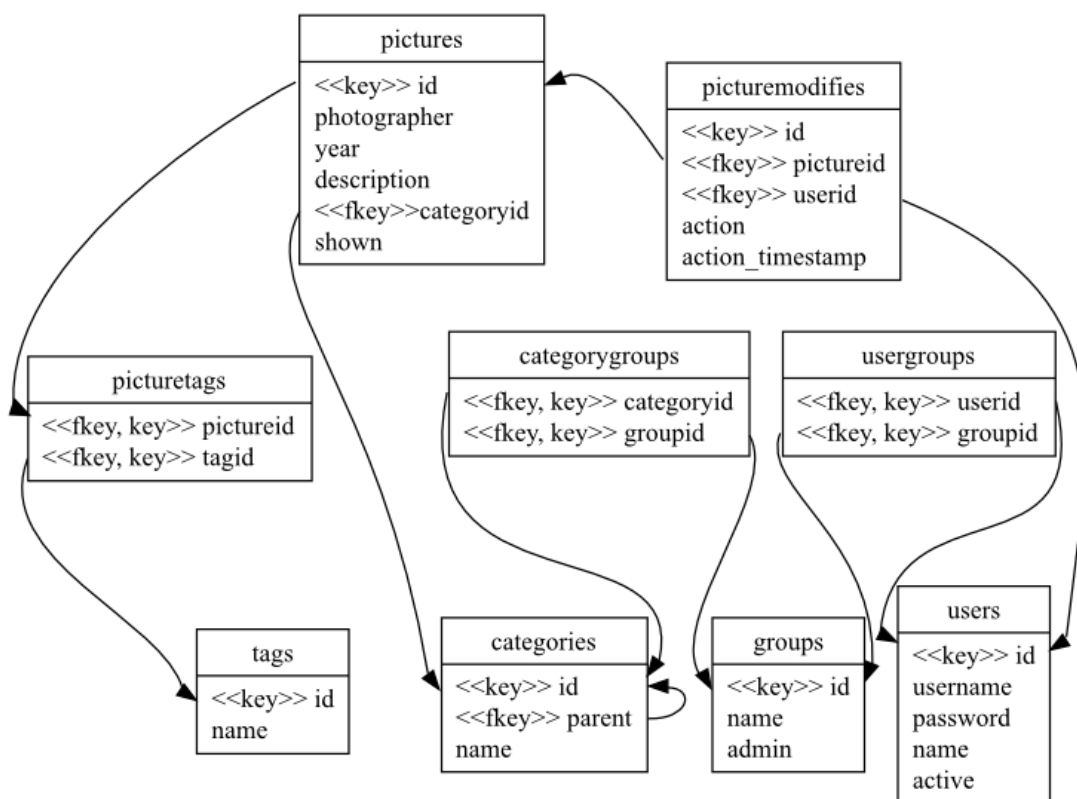


Kuva 4: Käyttöliittymäkaavio

## 6 Relaatiotietokanta

Relaatiotietokanta esitellään kahdella tavalla, ensin kaaviona ja sitten SQL-lauseilla.

### 6.1 Relaatiotietokantakaavio



Kuva 5: Relaatiotietokantakaavio

### 6.2 SQL-lauseet

Luodaan perus tietokanta Create Table -lauseilla ja erikseen näkymä, jotta tietokantaa olisi järkevä käsitellä.

#### 6.2.1 Create Table

```
1 CREATE SCHEMA galleria;
```

```
1 CREATE TABLE galleria.users (  
2 id serial PRIMARY KEY,  
3 username character varying NOT NULL UNIQUE  
  CHECK (char_length(username) > 0),  
4 password character varying NOT NULL CHECK  
  (char_length(password) > 0),  
5 name character varying NOT NULL CHECK  
  (char_length(name) > 0),  
6 active boolean DEFAULT TRUE  
7 );
```

```
1 CREATE TABLE galleria.groups (  
2 id serial PRIMARY KEY,  
3 name character varying NOT NULL UNIQUE  
  CHECK (char_length(name) > 0),  
4 admin boolean DEFAULT FALSE  
5 );
```

```
1 CREATE TABLE galleria.usergroups (  
2 userid integer NOT NULL REFERENCES galleria.users(id)  
  ON DELETE CASCADE ON UPDATE CASCADE,  
3 groupid integer NOT NULL REFERENCES galleria.groups(id)  
  ON DELETE CASCADE ON UPDATE CASCADE,  
4 PRIMARY KEY (userid, groupid)  
5 );
```

```
1 CREATE TABLE galleria.categories (  
2 id serial PRIMARY KEY,  
3 parent integer REFERENCES galleria.categories(id)  
  ON DELETE CASCADE ON UPDATE SET NULL,  
4 name character varying NOT NULL  
  CHECK (char_length(name) > 0),  
5 weight integer DEFAULT 0  
6 );
```



```

1 CREATE TABLE galleria.pictures (
2 id serial PRIMARY KEY,
3 photographer character varying NOT NULL
  CHECK (char_length(photographer) > 0),
4 year char(4) NOT NULL
  CHECK (char_length(year) = 4),
5 description character varying NOT NULL
  CHECK (char_length(description) > 0),
6 categoryid integer NOT NULL REFERENCES galleria.categories(id)
  ON DELETE RESTRICT ON UPDATE RESTRICT,
7 shown boolean DEFAULT TRUE
8 );

```

```

1 CREATE TABLE galleria.categorygroups (
2 categoryid integer NOT NULL REFERENCES galleria.categories(id)
  ON DELETE CASCADE ON UPDATE CASCADE,
3 groupid integer NOT NULL REFERENCES galleria.groups(id)
  ON DELETE CASCADE ON UPDATE CASCADE,
4 PRIMARY KEY (categoryid, groupid)
5 );

```

```

1 CREATE TABLE galleria.tags (
2 id serial PRIMARY KEY,
3 name character varying NOT NULL UNIQUE
  CHECK (char_length(name) > 0 AND char_length(name) <= 20)
4 );

```

```

1 CREATE TABLE galleria.picturetags (
2 pictureid integer NOT NULL REFERENCES galleria.pictures(id)
  ON DELETE CASCADE ON UPDATE CASCADE,
3 tagid integer NOT NULL REFERENCES galleria.tags(id)
  ON DELETE CASCADE ON UPDATE CASCADE,
4 PRIMARY KEY (pictureid, tagid)
5 );

```

```

1 CREATE TABLE galleria.picturemodifies (
2 id serial PRIMARY KEY,
3 pictureid integer NOT NULL REFERENCES galleria.pictures(id)
  ON DELETE CASCADE ON UPDATE CASCADE,
4 userid integer NOT NULL REFERENCES galleria.users(id)
  ON DELETE RESTRICT ON UPDATE RESTRICT,
5 action character varying NOT NULL
  CHECK (char_length(action) > 0),
6 action_timestamp timestamp DEFAULT NOW()
7 );

```

### 6.2.2 Create View

```

1 CREATE OR REPLACE VIEW galleria.picturedata AS
2 SELECT p.id AS pid,
3 p.photographer AS pg,
4 p.year AS year,
5 p.description AS desc,
6 p.shown as shown,
7 c.id AS cid,
8 c.name AS cname,
9 cg.groupid AS gid
10 FROM galleria.pictures AS p
11 LEFT JOIN galleria.categories AS c
12 ON c.id = p.categoryid
13 LEFT JOIN galleria.categorygroups AS cg
14 ON cg.categoryid = c.id
15 ORDER BY pid DESC;

```

```

1 CREATE OR REPLACE VIEW galleria.tagdata AS
2 SELECT t.id AS tid,
3 t.name AS name,
4 COUNT(t.id) AS weight,
5 (
6 SELECT count(*)
7 FROM galleria.picturetags
8 ) AS sum
9 FROM galleria.tags AS t
10 JOIN galleria.picturetags AS pt
11 ON pt.tagid = t.id
12 GROUP BY t.id,
13 t.name
14 ORDER BY name ASC;

```

```

1 CREATE OR REPLACE VIEW galleria.tagsofpictures AS
2 SELECT t.id AS tid,
3 t.name AS name,
4 pt.pictureid AS pid
5 FROM galleria.tags AS t
6 JOIN galleria.picturetags AS pt
7 ON pt.tagid = t.id
8 ORDER BY tid ASC;

```

```
1 CREATE OR REPLACE VIEW galleria.groupsofuser AS
2 SELECT ug.userid AS uid,
3        g.id AS gid,
4        g.name AS name,
5        g.admin AS adm
6 FROM galleria.groups AS g
7 JOIN galleria.usergroups AS ug
8 ON ug.groupid = g.id
9 ORDER BY uid;
```

```
1 CREATE OR REPLACE VIEW galleria.groupsofcategory AS
2 SELECT cg.categoryid AS cid,
3        g.id AS gid,
4        g.name AS name,
5        g.admin AS adm
6 FROM galleria.groups AS g
7 JOIN galleria.categorygroups AS cg
8 ON cg.groupid = g.id
9 ORDER BY cid;
```

```
1 CREATE OR REPLACE VIEW galleria.userdata AS
2 SELECT u.id AS uid,
3        u.username AS username,
4        u.name AS name,
5        u.active AS active,
6        ug.groupid AS gid
7 FROM galleria.users AS u
8 LEFT JOIN galleria.usergroups AS ug
9 ON ug.userid = u.id
10 ORDER BY uid;
```

```

1 CREATE OR REPLACE VIEW galleria.categorylist AS
2 SELECT c.id AS cid,
3 c.parent AS cpar,
4 c.name AS name,
5 (
6 SELECT count(*) > 0
7 FROM galleria.pictures AS p
8 WHERE p.categoryid = c.id
9 AND p.shown = TRUE
10 ) AS haspics,
11 cg.groupid AS gid
12 FROM galleria.categories AS c
13 LEFT JOIN galleria.categorygroups AS cg
14 ON cg.categoryid = c.id
15 ORDER BY cpar ASC, c.weight ASC;

```

```

1 CREATE OR REPLACE VIEW galleria.changelog AS
2 SELECT pm.pictureid AS pid,
3 u.name AS user,
4 pm.action AS action,
5 pm.action_timestamp AS timestamp
6 FROM galleria.picturemodifies AS pm
7 JOIN galleria.users AS u
8 ON u.id = pm.userid
9 ORDER BY timestamp ASC;

```