

# **Tietokantasovelluksen Toteutusdokumentti**

Antti Tupamäki

Helsinki 28.02.10

HELSINGIN YLIOPISTO  
Tietojenkäsittelytieteen laitos

HELSINGIN YLIOPISTO –HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta/Osasto –Fakultet/Sektion –Faculty/Section		Laitos –Institution –Department	
Matemaattis-luonnontieteellinen tiedekunta		Tietojenkäsittelytieteen laitos	
Tekijä – Författare –Author			
Antti Tupamäki			
Työn nimi – Arbetets titel –Title			
Tietokokantasovelluksen suunnitteludokumentti			
Oppiaine –Läroämne –Subject			
Tietojenkäsittelytiede			
Työn laji – Arbetets art –Level	Aika – Datum –Month and year	Sivumäärä – Sidoantal –Number of pages	
	28.02.10		
Tiivistelmä – Referat –Abstract			
Toteutusdokumentti			
Avainsanat – Nyckelord –Keywords			
Säilytyspaikka –Förvaringställe –Where deposited			
Muita tietoja –Övriga uppgifter –Additional information			

## Table of Contents

<b>1 Johdanto.....</b>	<b>1</b>
1.1 Rajauksia.....	1
<b>2 Ohjelmiston yleisrakenne.....</b>	<b>1</b>
<b>3 Järjestelmän komponentit.....</b>	<b>2</b>
Image-model.....	2
User-model.....	2
Comment-model.....	2
Users-controller.....	2
Images-controller.....	2
Log-controllerissa.....	3
Kualauta-controllerissa.....	3
Application-controller.....	3
Kualauta-view.....	3
Search-view.....	3
Login-view.....	3
newimage-view.....	3
New user view.....	4
Edit user view.....	4
Appllication helper .....	4
<b>4. Asennustiedot.....</b>	<b>4</b>
<b>5 Käyttöohjeet.....</b>	<b>4</b>
<b>6 liitteet.....</b>	<b>4</b>

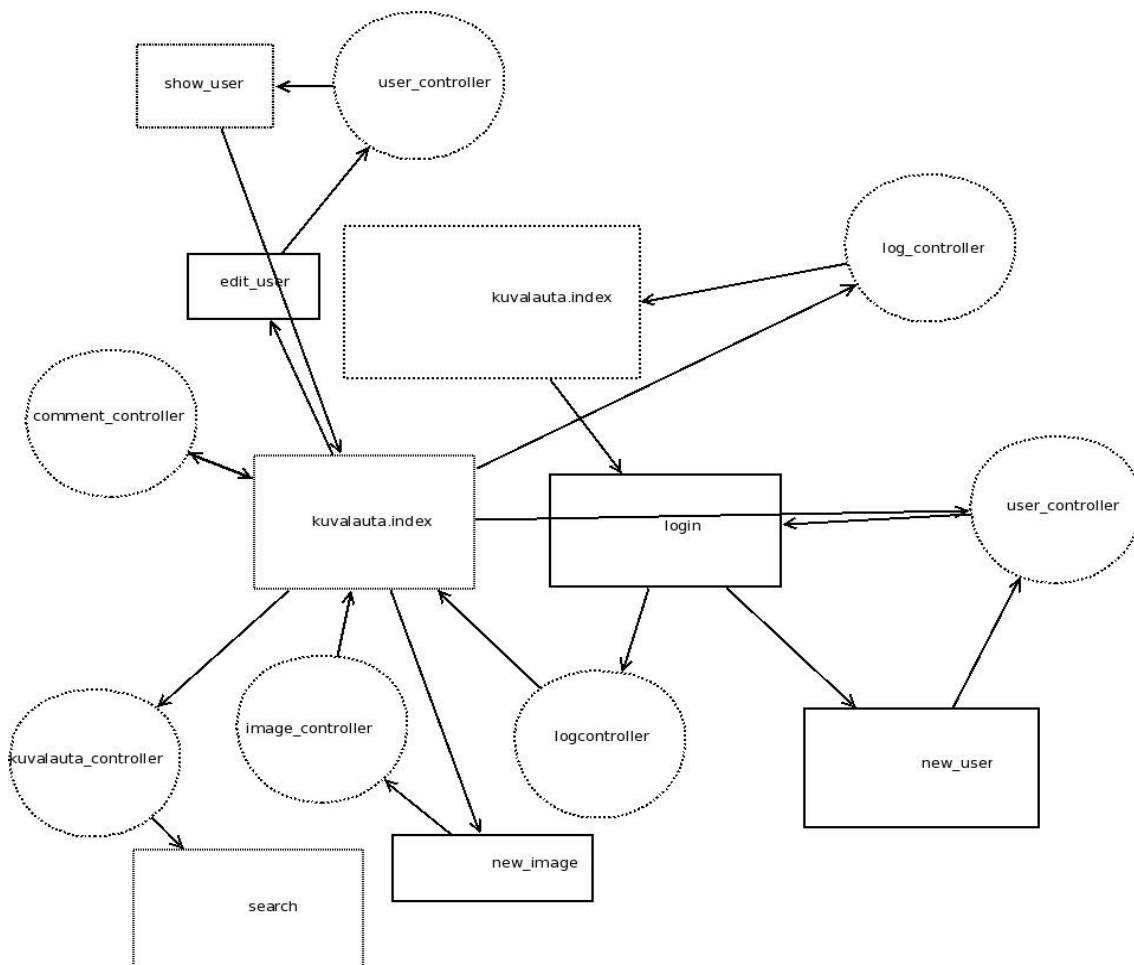
# 1 Johdanto

Työn aiheena on kuvagalleria johon voi lisätä ja poistaa kuvia. Käyttäjät voivat vapaasti myös kommentoida kuvia. Tietokannasta onnistuu myös kuvien haku kuvan nimellä. Järjestelmä on lähinnä tarkoitettu omien kuvien julkaisuun ja palautteen antamiseen. Järjestelmän tavoite on antaa helppokäyttöinen ja nopea tapa levittää kuvia. Sekä antaa mahdollisuus kommentoida näitä kuvia.

## 1.1 Rajauksia

En valitettavasti ehtinyt siivoamaan reititystä joten reitityksessä on paljon turhia reittejä. Searchissa ei ole nappia jota alunperin meinasin ja rekisteröinti ei ole aivan yhtä laajaa. Myös ajan puutteen takia en ehtinyt saada User-modellin validointia kuntoon.

## 2 Ohjelmiston yleisrakenne



### 3 Järjestelmän komponentit

Järjestelmä on tehty Ruby on the Railsilla joten, se jäljittelee perinteistä MVC-mallia. Tässä osassa luetellaan kaikki MVC-mallin osat.

#### Image-model

Tämä kuva kuuluu aina yhdelle käyttäjälle ja sillä voi olla monia kommentteja. Se sisältää tarpeelliset `named_scope`t joilla voidaan hakea kuvan lataajaa ja kuvia metodilla `search`. Model myös validoi kuvan, lataajan, ja nimen että nämä ovat oikeaa muotoa ja ovat olemassa. Kuvalla on myös jotain rajoituksia kuten että se ei saa olla suurempi kuin 2megatavua ja sen pitää olla muotoa `jpg/git/png`.

#### User-model

tämä malli validoi että `name`, `password` ja `email` ovat oikeaa muotoa. Tässä mallissa myös varmistetaan `Authenticate`-metodilla että käyttäjällä on oikeudet selata tiettyjä sivuja.

#### Comment-model

Kommentit kuuluvat yhdelle käyttäjälle ja kuvalle. Modellissa myös validoidaan, tarvittavat tiedot.

#### Users-controller

tässä kontrollerissa hoidetaan parametrien välitys user-modelille, mm käyttäjien luonti, editointi, näyttäminen, päivittäminen, ja tuhoaminen. Näistä kaikki muut paitsi uusien käyttäjien luonti on rajoitettu siten että käyttäjä pääsee ainoastaan muokkaamaan omia tietojaan.

#### Images-controller

Tässä kontrollerissa hoidetaan kuvien näyttäminen, luonti ja tuhoaminen. Ja ennen autentikoitumista kuvat voi ainoastaan nähdä sekä ainoastaan sellainen käyttäjä joka on

ladannut kuvan voi poistaa kyseisen kuvan.

## **Log-controllerissa**

hoidetaan sisäänkirjautuminen login-metodilla ja logout metodilla uloskirjautuminen.

## **Kuvalauta-controllerissa**

listataan kuvat index-metodilla, luodaan uusi kommentti comment-metodilla, etsitään search-metodilla ja saadaan etsinnän tulos show-results metodilla.

## **Application-controller**

tässä kontrollerissa on authenticate-metodi jolla tarkistetaan onko käyttäjä kirjautunut sisään.

## **Kuvalauta-view**

sisältää pääikkunan tässä ikkunassa, näytetään kommentit ja kuvat tähän ikkunaan voi myös jättää kommentteja.

## **Search-view**

tämä listaa kuvat ja kommentit samalla tavalla kuin kuvalauta view, mutta ei listaa kaikkia kuvia vaan ainoastaan hakutulokset.

## **Login-view**

sisältää sisäänkirjautumiseen vaadittavan lomakkeen ja kentät password ja username

## **newimage-view**

sisältää uuden kuvan lataamiseen vaadittavan lomakkeen, johon syötetään kuvan nimi, kuvan osoite koneella, ja kuitataan nämä.

### **New user view**

sisältää uuden käyttäjän luomiseen tarvittavat lomakkeen, nimi, salasana ja email joka pitää olla muodossa [jotain@jotain.fi](mailto:jotain@jotain.fi).

### **Edit user view**

tällä voi muuttaa käyttäjän tietoja. Heittää tämän jälkeen show user viewiin jossa nähdään muutetut tiedot.

### **Aplication helper**

sisältää user\_id kohdan joka palauttaa user\_id:sessioneista

## **4. Asennustiedot**

Ohjelman voi ”pullata” Git version hallintaa käyttäen serverille jossa, annetaan oikeudet lukea public html kansiota ja oikeudet lukea sekä kirjoittaa app kansion controller, view ja model:leja.

## **5 Käyttöohjeet**

Ohjelma on osoitteessa [anttutup.rails.cs.helsinki.fi](http://anttutup.rails.cs.helsinki.fi) ja kooodi on luettavissa <http://github.com/anttutup/tsoha> sivulla. Search:hiin en kerennyt laittaa nappia joten se toimii vain entteriä painamalla.

## **6 liitteet**