

Helsingin yliopisto, tietojenkäsittelytieteen laitos

Aineopintojen harjoitustyö: Tietokantasovellus

Gallery-site

<http://gallery.tuhoojabotti.com>

13.3.2014

014245291 Ville Lahdenvuo
ville.lahdenvuo@cs.helsinki.fi
<http://tuhoojabotti.com>

1. Sisällys

[1. Sisällys](#)

[2. Johdanto](#)

[3. Yleiskuva järjestelmästä](#)

[3.1. Käyttötapauskaavio](#)

[3.2. Käyttäjäryhmät](#)

[3.2.1. Jokamies](#)

[3.2.1. Kirjautunut](#)

[3.2.1. Ylläpitäjä](#)

[3.3. Käyttötapauskuvaukset](#)

[3.3.1. Kuvien tuominen](#)

[3.3.2. Kuvien katseleminen](#)

[3.3.3. Kuvien arvosteaminen](#)

[3.3.4. Kuvien hallinta](#)

[3.3.5. Kansioden luominen](#)

[3.3.6. Kansion poistaminen](#)

[4. Käyttöliittymä](#)

[4.1. Alustava sivukartta](#)

[4.2. Kirjautuminen](#)

[5. Tietosisältö](#)

[5.1. Alustava suunnitelma](#)

[5.2. Tietokohheet](#)

[5.2.1. Kansio](#)

[5.2.2. Kuva](#)

[5.2.3. Tägi](#)

[5.2.4. Käyttäjä](#)

[5.2.5. Arvostelu](#)

[5.3. Relaatiotietokantakaavio](#)

[6. Järjestelmän yleisrakenne](#)

[6.2. Palvelin](#)

[6.3. Frontend](#)

[7. Käyttöliittymä ja järjestelmän komponentit](#)

[8. Asennustiedot](#)

[9. Käyttöohje](#)

2. Johdanto

Tämä on dokumentaatio tietokantasovelluksen harjoitustyöstäni. Työn ensisijaisena tavoitteena on saada aikaan toimiva järjestelmä oman valokuvausharrastukseni tueksi. Tällä hetkellä kotisivuni tai portfolioni on aika suppea, sillä en saa töitäni esille hyvin. Olen yrittäny ratkaista ongelmaa ulkoisilla palveluilla tai erilaisilla itse asennettavilla järjestelmillä, mutta mikään ei ole vastannut vaatimuksiani tähän mennessä. Tarkoitus on saada helposti ja näyttävästi valokuviani esille, mutta myös turvallisesti jakaa niitä ihmisiille.

Sovellusta on tarkoitus ajaa Kapsi Internet-käyttäjät Ry:n palvelimilla ja toteutuskieleksi on valittu JavaScript. Tietokannaksi tulee MySQL, sillä sellainen minulla on jo. Todennäköisesti tietokannan vaihtaminen tulee olemaan melko helppoa, sillä sovelluksen rakenteesta tulee hyvin modulaarinen. Sovellus tulee vaatimaan selaimelta JavaScript-tuen.

Järjestelmä rakentuu erillisistä komponenteista, jotka toimivat yhdessä. Palvelin tarjoilee tarvittavat rajapinnat RESTful-periaatteiden mukaan ja frontend huolehtii sisällön näyttämisestä käyttäjälle. Toteutuksessa alustana on palvelinpuolella Node.js ja frontend tulee käyttämään AngularJS-sovelluskehystä.

3. Yleiskuva järjestelmästä

3.1. Käyttötapauskaavio



3.2. Käyttäjäryhmät

3.2.1. Jokamies

Jokainen voi katsella kuvia, paitsi niitä, jotka ovat suojaatuissa kansioissa.

3.2.1. Kirjautunut

Kirjautunut käyttäjä voi arvostella kuvia ja "tallentaa kansioiden salasanoja".

3.2.1. Ylläpitäjä

Ylläpitäjällä on oikeus lisätä, poistaa ja muokata kaikkea.

3.3. Käyttötapauskuvaukset

Minulla on tiettyjä ominaisuuksia, joita kuvagallerialta kaipaan. Tässä on lista tärkeimmistä, jotka tulen toteuttamaan kurssin aikana.

3.3.1. Kuvien tuominen

Tiedostojen uploadaaminen selaimella ei ole ensisijainen käyttötapaus, sillä kuvien lisäämisen ei tarvitse onnistua multa kuin minulta, ja minulle se on kätevintä sftp:llä.

Ylläpitäjän käyttöliittymässä on “import”-nappula, joka etsii tiedostojärjestelmästä asetukseen laitetusta sijainnista kuvia, joita ei ole vielä lisätty galleriaan.

Järjestelmä listaa uudet tiedostot ja antaa tallentaa niille metatietoja, kuten nimen, kansion, tägeja, jne. Esikatselukuvat luodaan automaattisesti ja napin painalluksella kuvat on lisätty galleriaan.

3.3.2. Kuvien katseleminen

Aika itsestäänselvyys. Kuvia pitää voida katsella selaimella, kansioita pitää voida navigoida ja kuvia pitää voida hakea tägien/nimen perusteella.

Mikäli kansio on salasanasuojattu, pitää ensin syöttää salasana, ellei ole jo istunnossa kerran syöttänyt. Jos salasanalla syöttää kirjautuneena, muistetaan se kyseiselle käyttäjälle tulevaisuudessa.

3.3.3. Kuvien arvosteaminen

Kirjautunut käyttäjä voi arvostella kuvia asteikolla 1-5, mahdollisesti myös lyhyen tekstimuotoisen palautteen voi antaa. Tämä tapahtuu yksittäisen kuvan sivulla tähtiarvostelulla.

3.3.4. Kuvien hallinta

Kaikkia kuvien tietoja voi muokata käyttöliittymästä. Tietoihin kuuluu nimi, tägit, kansio, exif-tiedot.

3.3.5. Kansioiden luominen

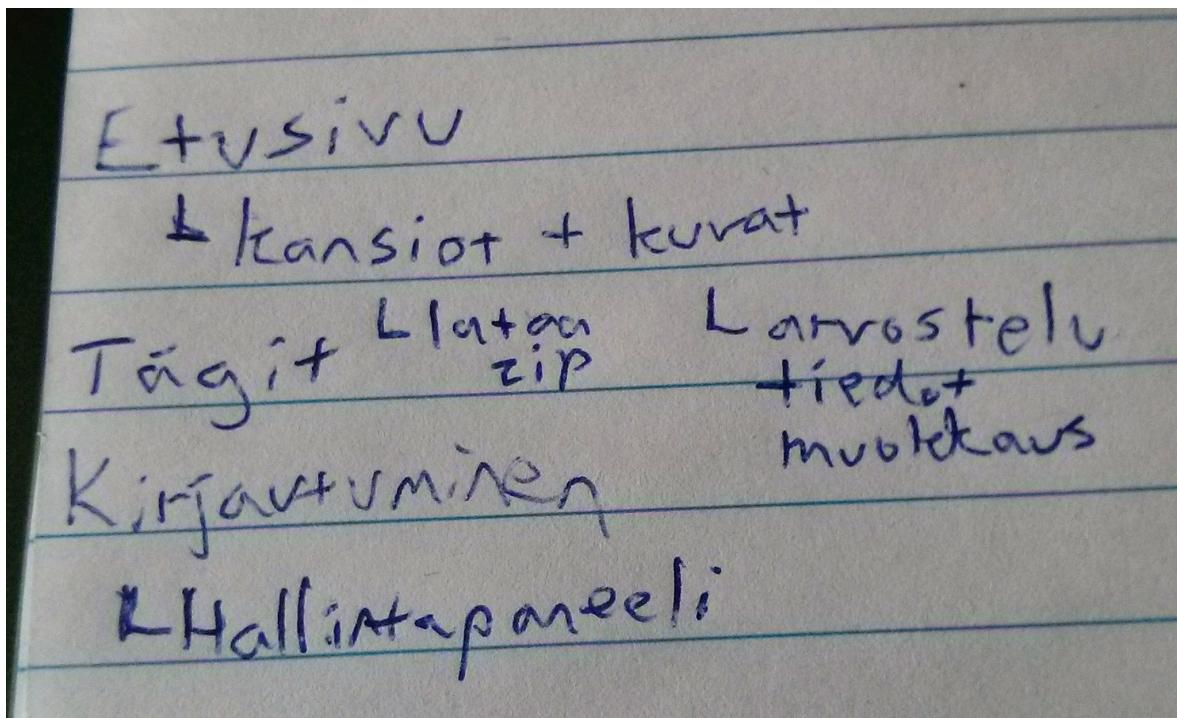
Kansioita voi luoda kansioiden sisään ja niille voi asettaa salasanan, jolla kansion saa yksityiseksi.

3.3.6. Kansion poistaminen

Kansioita voi poistaa, jolloin niissä olevat kuvat tuhoutuvat.

4. Käyttöliittymä

4.1. Alustava sivukartta



4.2. Kirjautuminen

Sivusto käyttää kirjautumiseen Facebookin rajapintaa. Idea on, että sivustoon ei tarvitse kirjautua ollenkaan, mutta jos haluaa arvostella kuvaaa, niin on kirjauduttava. Lisäksi kirjautumalla voi tallentaa oikeuden suojaan kansioon.

```
① ► POST https://secure.tuhoojabotti.com/gallery/user/verify 401 (Unauthorized) angular.js:8165
    show modal                                         main.js:72
    fb logged in!                                     login.js:19
    logged in!                                         main.js:78
    api verified! Object {id: "1247782261", name: "Ville Lahdenvuo", admin: 1} login.js:28
    fb logout                                         login.js:34
    login cancelled!                                  main.js:88
② ► POST https://secure.tuhoojabotti.com/gallery/user/verify 401 (Unauthorized) angular.js:8165
    show modal                                         main.js:72
    login cancelled!                                main.js:88
```

Kuten kuvasta nähdään, kun käyttäjä yrittää tehdä jotain, mikä vaatii autentikointia, palvelin vastaa tietenkin 401-virhekoodilla. Sovellus huomaa tämän ja näyttää kirjautumisnappulan. Kun käyttäjä on kirjautunut, Facebookin rajapinta antaa access tokenin, joka liitetään osaksi

jokaista tulevaa HTTP-pyyntöä. Sen takia kun sovellus yrittää kutsua uudelleen, se onnistuu. Mikäli kutsu tehdään validilla access tokenilla, mutta käyttäjää ei ole tietokannassa, käyttäjä lisätään tietokantaan.

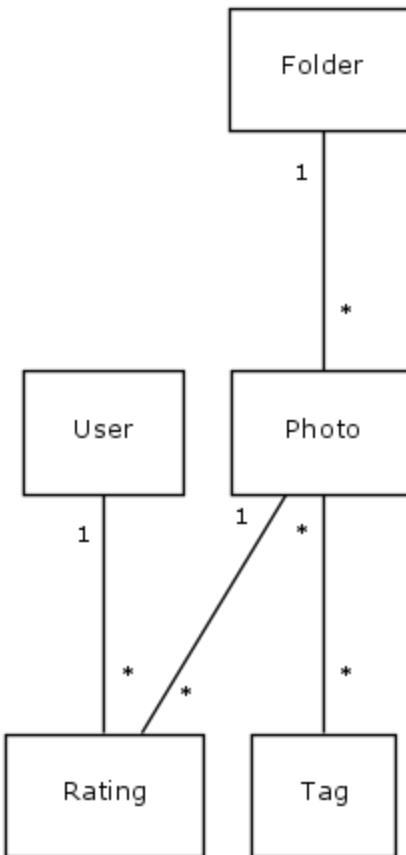
user/verify API on tarkoitettu vain kirjautumisen testaamiseen. Lopulliseen sovellukseen ei tule erillistä sisäänkirjautumistoimintoa, vain uloskirjautumiseen.

Verifiointi on helppo yhdistää järjestelmään:

```
6  get('/users', 'users#index');
7  post('/user/verify', verify, 'users#log');
8  get('/user/:id', 'users#show');
```

5. Tietosisältö

5.1. Alustava suunnitelma



5.2. Tietokohdeet

5.2.1. Kansio

Attribuutti	Arvojoukko	Kuvailu
Nimi	Merkkijono, max. 120 merkkiä	Kansion nimi
Kuvaus	Merkkijono, max. 1000 merkkiä	Kansion sisällön kuvaus
Salasana	Merkkijono, max. 120 merkkiä	Kansion salasana

5.2.2. Kuva

Attribuutti	Arvojoukko	Kuvailu
Nimi	Merkkijono, max. 120 merkkiä	Kuvan nimi
Polku	Merkkijono, max. 120 merkkiä	Kuvan polku
Kuvaus	Merkkijono, max. 1000 merkkiä	Kuvan kuvaus

5.2.3. Tägi

Attribuutti	Arvojoukko	Kuvailu
Nimi	Merkkijono, max. 120 merkkiä	Tägin nimi
Kuvaus	Merkkijono, max. 1000 merkkiä	Tägin kuvaus

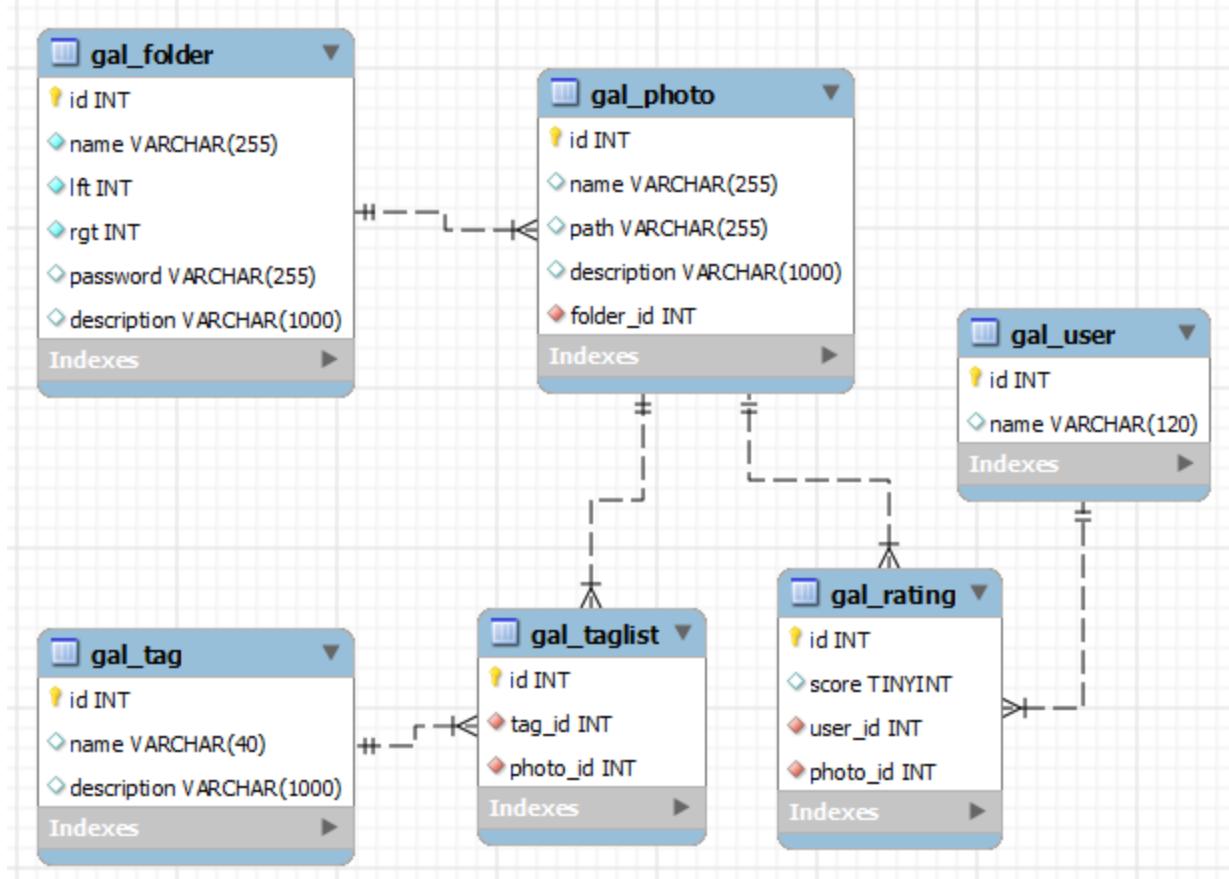
5.2.4. Käyttäjä

Attribuutti	Arvojoukko	Kuvailu
Nimi	Merkkijono, max. 120 merkkiä	Käyttäjänimi

5.2.5. Arvostelu

Attribuutti	Arvojoukko	Kuvailu
Pisteet	Kokonaisluku 1-5	Tähtiavostelu

5.3. Relaatiotietokantakaavio



6. Järjestelmän yleisrakenne

Sovellus käyttää tavallaan MVC-mallia. Sovelluksen data, logiikka ja näkymät on eroteltu järkevästi toisistaan.

6.2. Palvelin

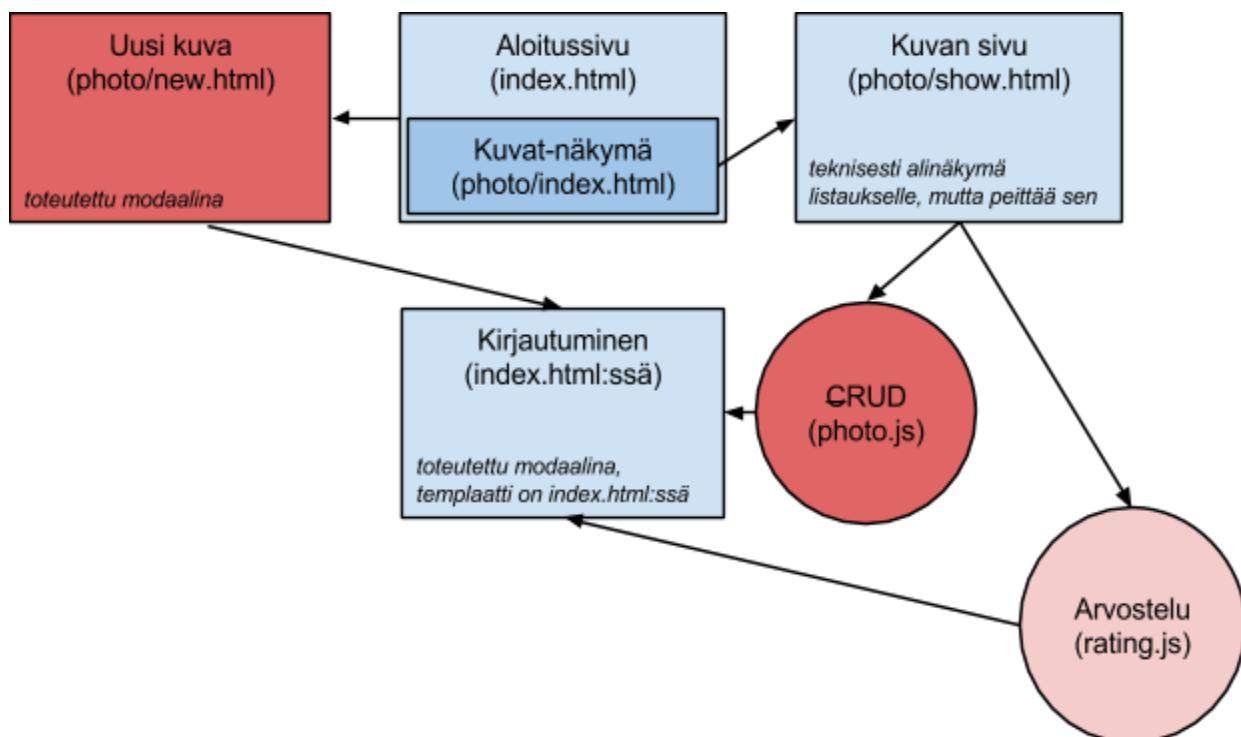
Palvelin löytyy kansiosista "api" ja se on vain REST-rajapinta. Mallit löytyvät "models" ja kontrollerit "controllers" kansioista. Reititys hoidetaan "routes"-tiedostolla. Apukirjastoja on kaksi, "activerecord"-moduuli on tietokanta abstraktio ja "accesstoken"-moduuli toimii autentikaatio middlewarearena. "index.js"-tiedosto kasaa kaikki moduulit yhdeksi toimivaksi kokonaisuudeksi. Yläkansiossa oleva "app.js"-tiedosto vain hoitaa riippuvuuksien lataamisen ja injektoimisen. Sovelluksen asetukset tulee ".env"-tiedostoon.

6.3. Frontend

Frontend puoli löytyy kansiosta "www". Kaikki pyynnöt ohjataan ".htaccess"-tiedostolla "index.html"-sivulle ja reititys tapahtuu javascriptillä ja HTML5 history-apilla. Näkymät on sijoitettu "views"-kansioon ja sovelluksen osia hoitavat javascript-tiedostot ovat "js"-kansiossa ja ne on nimetty kuvaavasti. Tiedostoja voisi myös jakaa pienempiin komponentteihin, mutta sitten tulisi vielä tärkeämäksi saada mukaan tiedostojen minifiointi ynnä muuta. Sovelluksen asetukset (apin url, html5-mode) ovat "main.js"-tiedostossa.

Koodin kommentointi on vielä työn alla.

7. Käyttöliittymä ja järjestelmän komponentit



Legend: Punaisella kirjautumista vaativat operaatiot, tummemmalla vain adminille.

Sovelluksen rakenne on vähän monimutkainen, koska näytöjä ei ole montaa, vaan kaikki tapahtuu javascriptillä ja ajaxilla. Esimerkiksi arvostelu annetaan vain painamalla tähtiä kuvan sivulla. Tägeille tulee vielä oma sivunsa.

8. Asennustiedot

Sovelluksen asennus on helppoa. Asennetaan Node.js ja MySQL. Kloonataan projektin tiedostot ja ajetaan npm install -komento, joka asentaa projektin riippuvuudet.

Projektin asetukset ovat .env-nimisessä tiedostossa, johon tulee kaikki asetukset tietokannasta HTTP-porttiin.

Tarvittavat asetukset ovat

- MYSQL_SERVER (serverin osoite)
- MYSQL_USER (käyttäjätunnus)
- MYSQL_PASSWORD (salasana)
- MYSQL_DATABASE (tietokanta)
- MYSQL_TABLE_PREFIX (taululle annettu prefiksi)
- PORT (portti, jota REST API kuuntelee)

www-kansion sisältö kopioidaan jonnekin missä se palvelaan käyttäjälle esimerkiksi Apachella.

Tämän jälkeen vain suoritetaan forever app.js, joka käynnistää palvelimen ja pitää sen päällä, jos se kaatuu.

9. Käyttöohje

Aava projekti osoitteesta <http://gallery.tuhoojabotti.com/>.

Jos sinun tarvitsee kirjautua sisään käytä seuraavaa tunnusta:

- Tsoha Testerman: tsoha_jimbbkg_testerman@tfbnw.net - hassupassu

Sovelluksen pitäisi olla aika helppokäyttöinen.