

# BÄNDIREKISTERI

Juuso Saranki

TIETOKANTASOVELLUS, Harjoitustyö



## Sisällysluettelo

Johdanto .....	4
Käyttäjärühmät / käyttötapaukset .....	5
Käyttötapauskaavio.....	6
Järjestelmän tietosisältö .....	7
Käsitekaavio.....	7
Tietokohteet .....	7
Relaatiotietokantakaavio .....	10
Asennusohje .....	11
Käynnistys- / käyttöohje .....	12
Järjestelmän yleisrakenne .....	13
Kontrollerit ja niiden vastualueet: .....	13
Mallit .....	13
Näkymät .....	13
Käyttöliittymä- ja komponenttikaavio .....	14
Testaus, tunnetut bugit ja puutteet & jatkokehitysideat.....	15
Omat kokemukset .....	15

## Johdanto

Tämän järjestelmän tarkoitus on tarjota Bändeille oma ns. näyteikkuna, jossa käyttäjät voivat käydä arvostelemassa bändejä ja bändit voivat nostaa omaa sijoitustaan listalla korkeammalle sen mukaan, minkälaisia arvioita käyttäjät ovat antaneet. Sovellus on siis tehty tuomaan yhtyeille näkyvyyttä, sekä faneille mahdollisuus tukea suosikkibändiensä toimintaa.

Järjestelmän tavoite on olla mahdollisimman yksinkertainen, sekä tehokas käyttää. Tarkoitus on välttää ylimääräisiä klikkauksia, sekä ylimääräisiä sarakkeita/tauluja tietokannassa.

Sovellus toteutetaan käyttämällä Helsingin yliopiston users-palvelimen tsoha bootstrappia, mikä koostuu postgre sql –tietokannasta, PHP:sta, sekä html front endistä. Jos sovellusta siirretään toiselle alustalle, vaaditaan siis postgresql-tietokantasovellus, sekä PHP-tuki ja jokin web-palvelin, esim. Apache tai nginx.

## Käyttäjärühmät / käyttötapaukset

### **Rekisteröimätön käyttäjä**

Rekisteröimätön käyttäjä pääsee selailemaan bändien tietoja ja lukemaan arvosteluja, mutta ei voi itse luoda/muokata bändejä, eikä kirjoittaa arvosteluja.

Käyttötapausesimerkki: Käyttäjä selailee listalta bändejä, klikkaa bändin infosivun auki ja lukee bändille kirjoitetut arvostelut.

### **Rekisteröitynyt käyttäjä**

Rekisteröity käyttäjä pääsee selailemaan bändien tietoja ja lukemaan arvosteluja, sekä voi kirjoittaa arvosteluja. Rekisteröityneellä käyttäjällä on myös mahdollisuus luoda oma bändi, jolloin hänestä tulee bändin jäsen.

Käyttötapausesimerkki: Rekisteröitynyt käyttäjä löytyy palvelusta suosikkibändinsä ja haluaa kirjoittaa sille arvostelun. Käyttäjä painaa ”arvioi”-painiketta ja hänelle aukeaa arviointilomake, mihin käyttäjä määrittää arvosanan ja suullisen arvion bändille.

### **(Vähintään yhden) Bändin jäsen**

Rekisteröityneestä jäsenestä tulee bändin jäsen siinä vaiheessa kun hän joko luo itse uuden bändin palveluun, tai jonkun bändin jo olemassaoleva jäsen lisää käyttäjän bändiinsä jäseneksi. Samat käyttöoikeudet kuin rekisteröityneellä käyttäjällä, mutta voi lisäksi muokata oman bändinsä tietoja.

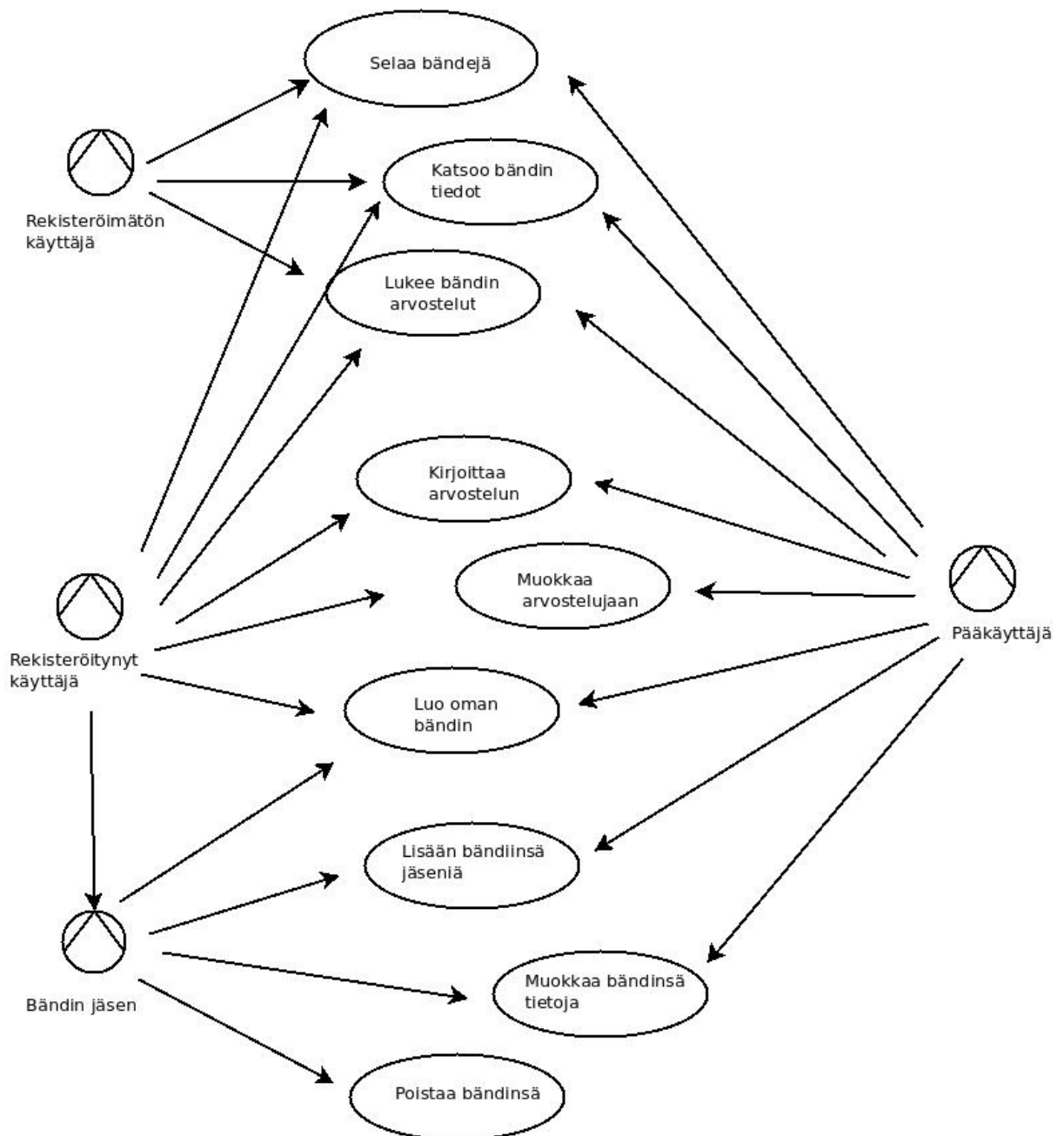
Käyttötapausesimerkki: Rekisteröitynyt käyttäjä päättää lisätä oman bändinsä palveluun ja hänestä tulee täten bändin jäsen. Rekisteröityneelle käyttäjälle tulee omat bändit – näkymään lista omista bändeistä, sekä mahdollisuus muokata bändi(e)n tietoja, sekä lisätä jäseniä bänd(e)i(h)in(sä).

### **Pääkäyttäjä**

Kyseistä hahmoa ei vielä järjestelmässä ole, mutta tarkoitus on myöhemmin lisätä vielä käyttäjärühmä, jolla on täydet oikeudet kaikkien arvioiden, sekä bändien ja käyttäjien muokkaamiseen, sekä poistamiseen.

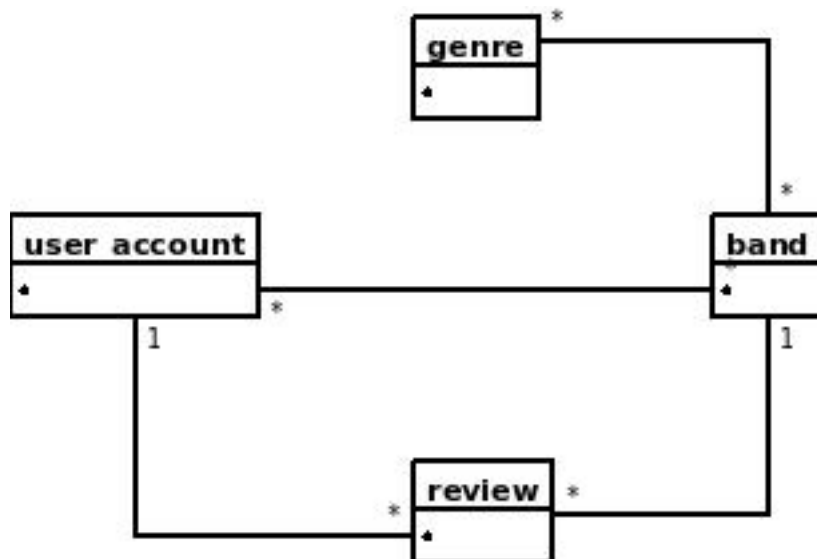
Käyttötapausesimerkki: Pääkäyttäjä poistaa kaikki bändit ja käyttäjät palvelusta ja saa kiitokseksi potkut työpaikastaan.

## Käyttötapauskaavio



## Järjestelmän tietosisältö

### Käsitekaavio



### Tietokohteet

Tietokohde: user\_account

Attribuutti	Arvojoukko	Kuvailu
User_accountid	Integer, 9 merkkiä, not null, auto incremental	Yhden rivin uniikiksi tekevä identifiointi-ID
username	Varchar, 200 merkkiä	Käyttäjätunnus
email	Varchar, 200 merkkiä	Käyttäjän sähköpostiosoite
password	Varchar, 200 merkkiä	Käyttäjän salasana
usertype	Integer, 1 merkki	Käyttäjän tyyppi (0= peruskäyttäjä, 1 = soittaja, 2 = admin)
firstname	Varchar, 200 merkkiä	Käyttäjän etunimi
lastname	Varchar, 200 merkkiä	Käyttäjän sukunimi
description	Varchar, 500 merkkiä	Käyttäjän kuvaus (tämä täytetään vain, jos kyseessä soittaja)

Tähän kohteeseen on tallennettu käyttäjien tiedot. Käyttäjillä on oikeus kirjoittaa arvosteluja bändeille. Yhteys siis reviews-tauluun. Jos käyttäjä on määritetty soittajaksi johonkin bändiin, on käyttäjällä reviews-taulun lisäksi yhteys userband-tauluun, jolla määritetään myös käyttäjän oikeudet kunkin bändin tietojen muokkaamiseen.

Tietokohde: band

Attribuutti	Arvojoukko	Kuvailu
bandID	Integer, 9 merkkiä, not null, auto incremental	Yhden rivin uniikiksi tekevä identifiointi-ID
Bandname	Varchar, 200 merkkiä	Bändin nimi
description	Varchar, 200 merkkiä	Bändin kuvaus
Bandtypeid	Int	Bändin tyyppi
Bandimagepath	Varchar, 500 merkkiä	Tähän tallennetaan URL mistä haetaan bändin kuva bändisivulle. Esim. Dropbox:n public jaolla kuva. (ei vielä käytössä.)
Established	integer	Bändin perustamisvuosi
Homecity	Varchar(400)	Bändin kotikaupunki
Country	Varchar(400)	Bändin kotimaa
Linktohomepage	Varchar(1000)	URL bändin kotisivulle
Twitter	Varchar(1000)	URL bändin twitter-sivulle
Facebook	Varchar(1000)	URL bändin facebook-sivulle
myspace	Varchar(1000)	URL bändin myspace-sivulle
youtubef	Varchar(1000)	URL bändin youtube-sivulle
soundcloud	Varchar(1000)	URL bändin soundcloud-sivulle

Bands-tauluun on tallennettu bändien tiedot. Taulusta on yhteys kaikkiin muihin tauluihin.

Tietokohde: userband

Attribuutti	Arvojoukko	Kuvailu
user_accountid	Integer, 9 merkkiä, not null, serial	User_account taulun foreign key
bandid	Integer, 9 merkkiä, not null, serial	Band-taulun foreign key

Userband on user\_account- ja band-taulujen liitostaulu. Primary key user\_accountid:n ja bandid:n yhdistelmä.

Tietokohde: genre

Attribuutti	Arvojoukko	Kuvailu
genreID	SERIAL Primary Key	Yhden rivin uniikiksi tekevä identifiointi-ID
Name	Varchar(4000)	Genren nimi

Genre-taulu sisältää listan valittavista genreistä.

Tietokohde: bandgenre

Attribuutti	Arvojoukko	Kuvailu
Bandid	Int	Band-taulun foreign key
Genreid	Int	Genre-taulun foreign key

Bandgenre on band- ja genre-taulun liitostaulu

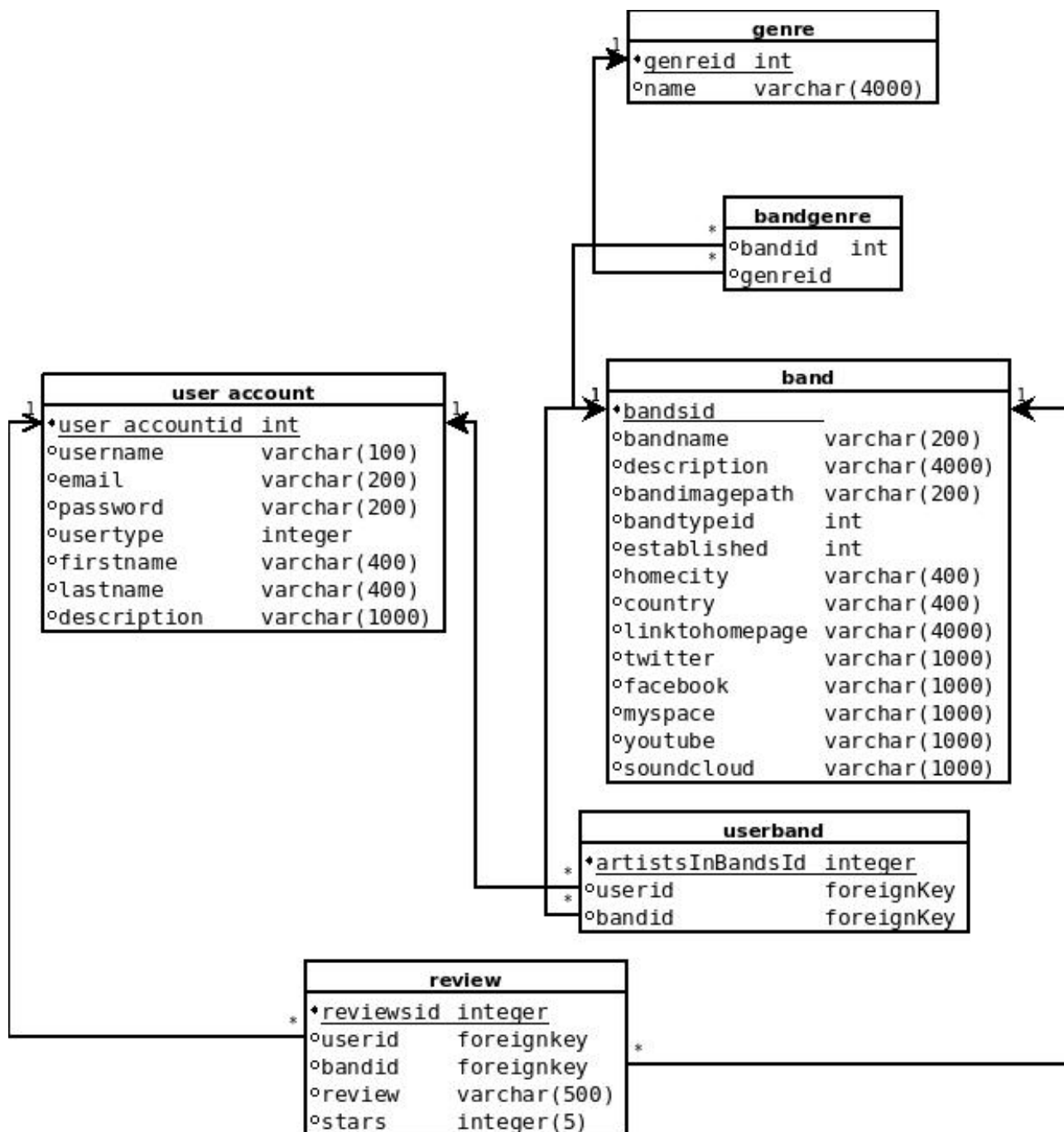


Tietokohde: review

Attribuutti	Arvojoukko	Kuvailu
reviewsID	Integer, 9 merkkiä, not null, auto incremental	Yhden rivin uniikiksi tekevä identifiointi-ID
userid	Foreign Key users-tilukosta	
bandid	Foreign Key bands-tilukosta	
review	Varchar, 4000 merkkiä	Tähän tallennetaan käyttäjän kirjoittama arvostelu yhteestä
stars	Integer, 1 merkki	Tähän tallennetaan arvosana (1-5), jonka käyttäjä antaa bändille.

Reviews-tiluun tallennetaan käyttäjien kirjoittamat arvostelut. Reviews-tiluun voi lisätä uusia rivejä ainoastaan rekisteröityneet käyttäjät.

## Relaattiotietokantakaavio



## Asennusohje

Sovelluksen saat otettua käyttöön siten, että kopioit kaikki tiedostot Tsoha-bootstrap-kansiosta haluamallesi webserverille. Aja ja sen jälkeen sql-kansiosta löytyvä create\_tables.sql-tiedosto. Voit myös halutessasi ajaa add\_test\_data.sql –tiedoston lisätäksesi testidataa palveluun. Suositus on kuitenkin, että lisäät palveluun alkuun vain yhden käyttäjätunnuksen, jolla voit sitten lisätä palveluun oman bändin sovelluksen käyttöliittymästä.

Suositus on käyttää palvelussa postgresql-tietokantaa, koska et saa sovellusta toimimaan esim. Mysql:n kanssa ilman, että muokkaat kaikki sql-queryt eri muotoon.

## Käynnistys- / käyttöohje

Sovellukseen pääsee kirjoittamalla internetselaimeen osoitteeksi

<https://sarankij.users.cs.helsinki.fi/tsoha/>

Tässä vaiheessa projektia palveluun kirjautuminen tapahtuu vielä osoitteessa

<https://sarankij.users.cs.helsinki.fi/tsoha/login>

(Kirjautumiseen tulee projektin valmistuttua myös linkkipainike yläpalkiin.)

Voit kirjatua palveluun

käyttäjätunnuksella: [john@usa.com](mailto:john@usa.com)

ja salasanalla: Kalle123

Toinen testitunnus millä voi testata käyttöoikeuksia:

Käyttäjätunnus: [Pertti.pasanen@hotmail.com](mailto:Pertti.pasanen@hotmail.com)

Salasana: Kalle123

Etusivulle aukeaa lista palveluun lisätyistä bändeistä.

Kun klikkaa bändin nimeä listalta, pääset selaamaan tarkempia tietoja bändistä.

Jos olet kirjautunut sisään, pääset myös muokkaamaan omien bändiesi tietoja, sekä lisäämään uusia bändejä ja poistamaan olemassaolevia (itse lisäämiäsi). Pystyt myös kirjoittamaan arvosteluja, sekä muokkaamaan / poistamaan aiemmin kirjoittamiasi arvioita. Näitä varten ilmestyy painikkeet vain jos on kirjautunut sisään.

Ilman kirjautumista voi palvelussa selata bändejä, mutta jos haluaa tehdä muutoksia bändien tietoihin tai kirjoittaa arvosteluja, tulee ensin kirjautua sisään.

## Järjestelmän yleisrakenne

Järjestelmä on toteutettu MVC-mallia noudattamalla. MVC:n mukaiset elementit sijaitsevat loogisesti niitä vastaavissa kansioissa. Kontrollerit löytyvät kansioista app/controllers, näkymät löytyvät kansioista app/views ja mallit löytyvät kansioista app/models. Käytetyt apukirjastot löytyvät kansioista lib. Reitit löytyvät config/routes.php –tiedostosta. Näkymien generoimiseen on käytetty apuna Twig-sivupohjamoottoria. Näkymissä käytettävä yläpalkki löytyy tiedostosta app/views/base.html. Kaikki tiedostonimet on kirjoitettu pienellä. Järjestelmä keskustelee postgresql:n kanssa PDO:ta käyttäen.

### Kontrollerit ja niiden vastuualueet:

#### **bandcontroller.php**

Vastaa pääsääntöisesti kaikesta bändiin liittyvästä listauksesta, muokkaamisesta, näyttämisestä jne.

#### **user\_controller.php**

Sisältää kirjautumiseen liittyvät toiminnot.

#### **reviewcontroller.php**

Sisältää bändien arvosteluihin liittyvät toiminnot (lisääminen, listaaminen, muokkaaminen ym.)

#### **bandmembercontroller.php**

Sisältää bändin jäsenien muokkaamiseen liittyvät toiminnot.

### Mallit

Mallit hoitavat yhteyden tietokantaan vastaavan nimisen kontrollerin tukena.

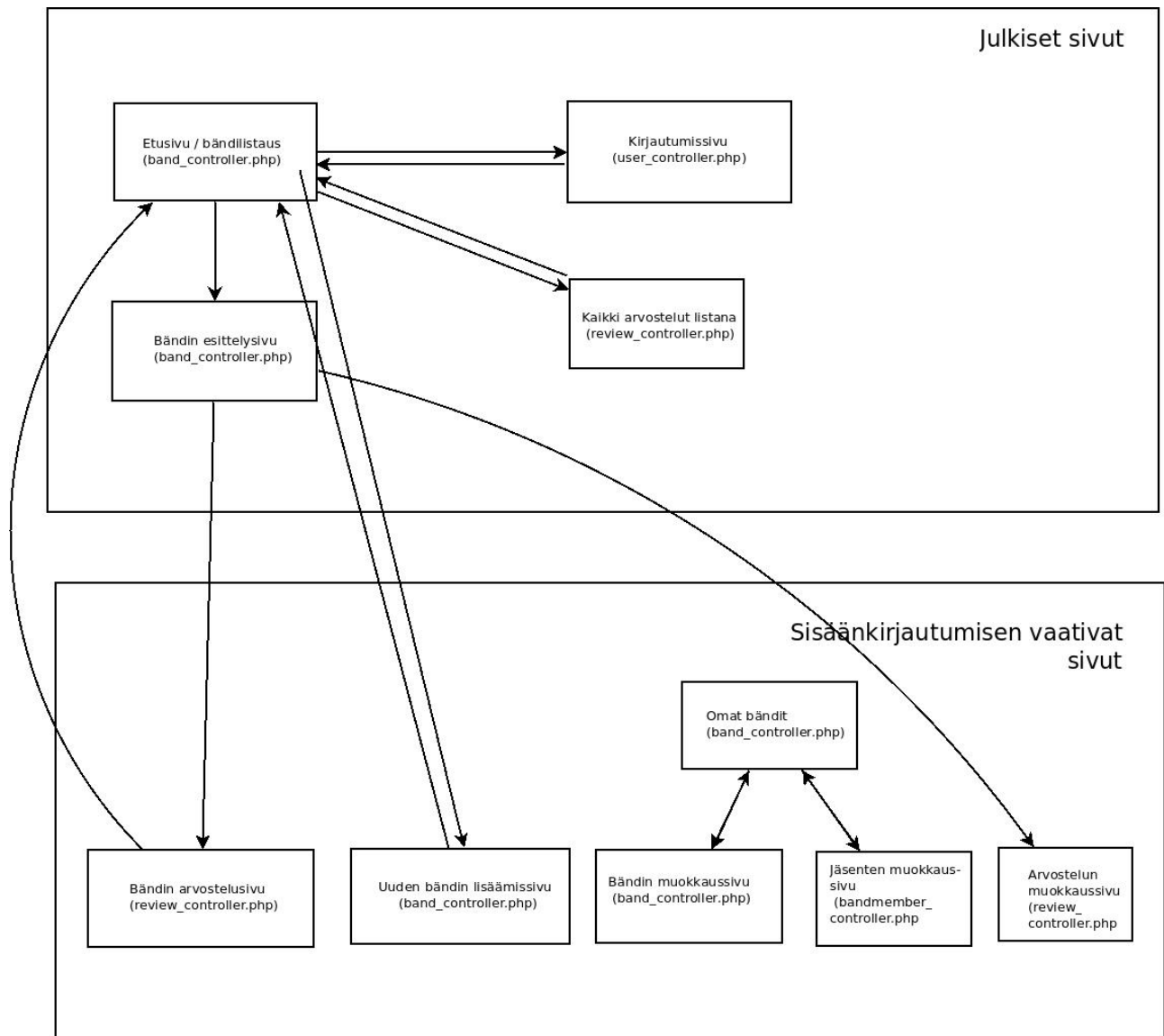
Pääasiassa mallit ovat nimetty sillä perusteella minkä tietokantataulun kanssa ne keskustelevat.

Poikkeuksena tästä bandmember, mikä toimii liitoksena bands, user\_account ja userband-taulujen välillä.

### Näkymät

Näkymät on pyritty nimeämään mahdollisimman kuvaavasti ja ne on jaettu kansioihin sen mukaan mihin osaan sovellusta ne liittyvät. Esim. Bändien listaamiseen, muokkaamiseen ym. Liittyvät näkymät löytyvät band-kansioista, kirjautumissivu löytyy user-kansioista ja arvioiden tekemiseen liittyvät asiat löytyvät review-nimisestä kansioista.

## Käyttöliittymä- ja komponenttikaavio



## Testaus, tunnetut bugit ja puutteet & jatkokehitysideat

Bugina ainakin se, että bändiä muokattaessa genre-listaan ei haeta bändin genrejä. Myöskään tallennettaessa bändin tiedot, ei poisteta genrejä, jos ne eivät enää ole muokkauksen jälkeen valittuna. Tämän korjaamiseen ei valitettavasti jäänyt enää muilta viimeistelyiltä aikaa.

Jatkokehitysideoina käyttäjienhallinta, sekä admin-käyttäjä. Tällä hetkellä käyttäjät lisätään palvelimelta käsin sql-komentoina. Genrejen ajatuksena oli tehdä niistä hakukenttä, johon voi kirjoittaa genren nimestä pari alkukirjainta ja kenttä täydentää loput kirjaimet. Tämän jälkeen genrejä sitten voisi valita esim. 5 per bändi ja bändeistä voisi kerätä sitten dataa, vaikka idealla "listaa kaikki bändit, joiden genre on heavy metal".

## Omat kokemukset

Koodaaminen on kivaa! Jos olisi ollut paremmin aikaa, niin olisin halunnut vielä monen monta muuta asiaa toteuttaa sovellukseen. Ajatus on kehittää sovellusta eteenpäin (graafinen ulkoasu ym.) heti kun vaan vapaata aikaa löytyy jostain.