

# Johdanto

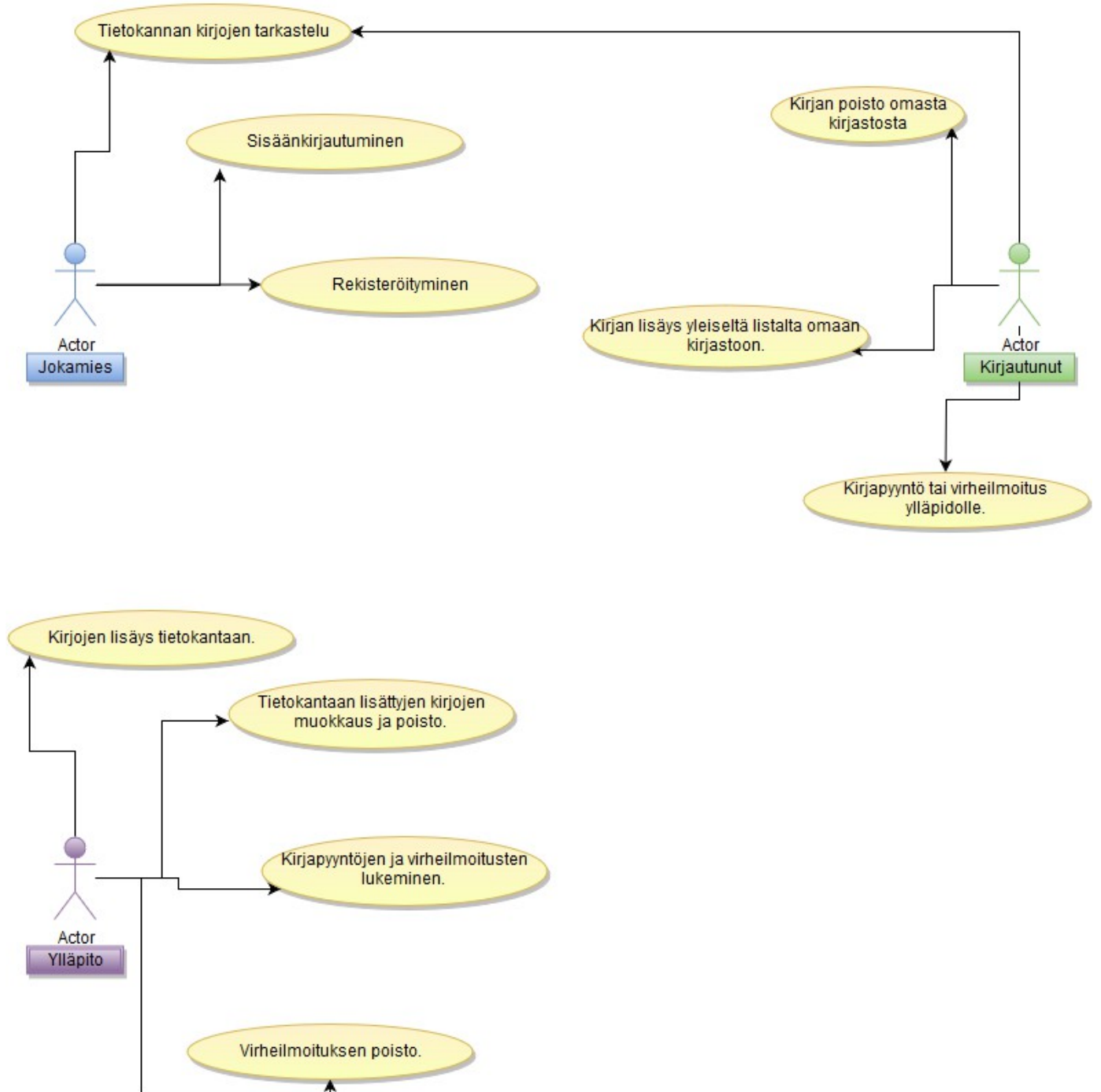
## Kirjatietokanta

Sovelluksen ideana on, että käyttäjät voivat listata mitä kaikkia kirjoja he ovat elämänsä aikana lukeeet. Jokaisella käyttäjällä on oma kirjasto, mihin he voivat lisätä kirjoja sovelluksen kirjalistasta.

## Tekniset tiedot

Työ toteutetaan laitoksen users-palvelimella, käytettävä tietokanta on PostgreSQL. Ohjelmointikielenä käytetään PHP:tä.

# Käyttötapauskaavio



## Käyttäjäryhmät:

### Jokamies

Jokamiehellä tarkoitetaan kaikkia kirjautumattomia henkilöitä sivulla.

### Kirjautunut

Kirjautunut käyttäjä on järjestelmään rekisteröitynyt käyttäjä.

### Ylläpito

Ylläpitoa on vain järjestelmän luonut käyttäjä, joka kirjautuu sisään omilla admin tunnuksilla.

## Käyttötapauskuvaukset:

### Jokamiehen käyttötapauskset

Tietokannan kirjojen tarkastelu:

Kuka tahansa voi tarkastella tietokannassa olevia kirjoja. Kirjautumatonkin käyttäjä näkee listan tietokannan kirjoista, ja voi tarkastella lisäksi jokaista kirjaa tarkemmin, jolloin näkee myös mahdollisen kirjalle asetetun esittelyn.

Muita käyttötapauskset: rekisteröityminen ja kirjautuminen

### Kirjautuneen käyttötapauskset

Kirjojen lisäys yleiseltä listalta omaan kirjastoon:

Käyttäjä voi lisätä kirjoja yleiseltä listalta omaan kirjastoonsa.

Kirjapyyntö tai virheimoitus ylläpidolle: (toteutettava vain jos aikaa jää)

Jos käyttäjä huomaa, että joku kirja puuttuu tietokannasta, hän voi lähettää ylläpidolle kirjapyynnön kirjan lisäämiseksi tietokantaan. Samaisella lomakkeella käyttäjä voi myös ilmoittaa ylläpidolle jossain kirjassa tai sovelluksen osassa olevasta virheestä.

Muita käyttötapauskset:

- Tietokannan kirjojen tarkastelu.
- Kirjan poistaminen omalta listalta.

## Ylläpidon käyttötapaukset

Kirjojen lisäys tietokantaan:

Ylläpito voi lisätä kirjoja tietokantaan, jolloin kirjasta pitää kertoa ainakin kirjan nimi ja kirjoittaja.

Tietokantaan lisättyjen kirjojen muokkaus ja poisto:

Ylläpito voi muokata kirjan tietoja, tai poistaa kirjoja kokonaan tietokannasta.

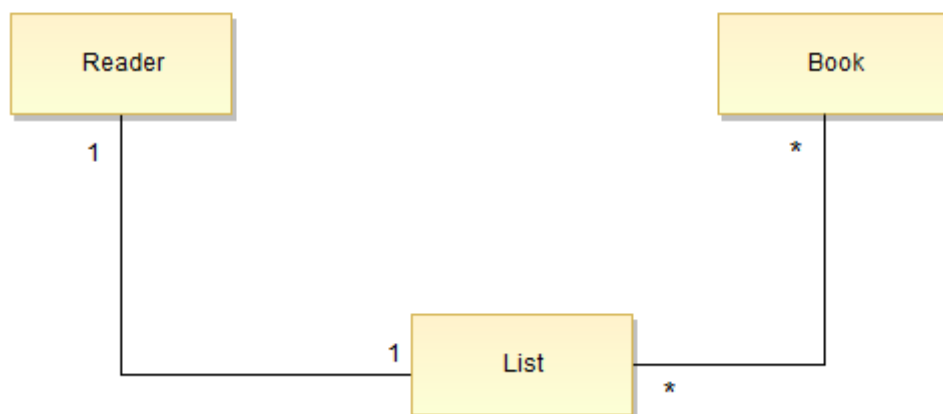
Kirjapyyntöjen ja virheilmoitusten lukeminen: (toteutettava vain jos aikaa jää)

Käyttäjien kirjoittamat kirjapyynnot ja virheilmoitukset tuleva ylläpidon nähtäville.

Virheilmoituksen poisto: (toteutettava vain jos aikaa jää)

Ylläpito voi poistaa jo toteutetut kirjapyynnot tai virheilmoitukset.

## Käsitekaavio



### **Tietokohde: Reader**

<b>Attribuutti:</b>	<b>Arvojoukko:</b>
Käyttäjätunnus	Merkkijono, max. 30 merkkiä
Salasana	Merkkijono, max. 50 merkkiä

**Kuvaus:** "Reader" kuvaa kirjautunutta käyttäjää, jolla on oma lista luetuile kirjoille.

### **Tietokohde: List**

<b>Attribuutti:</b>	<b>Arvojoukko:</b>
Ei ole	

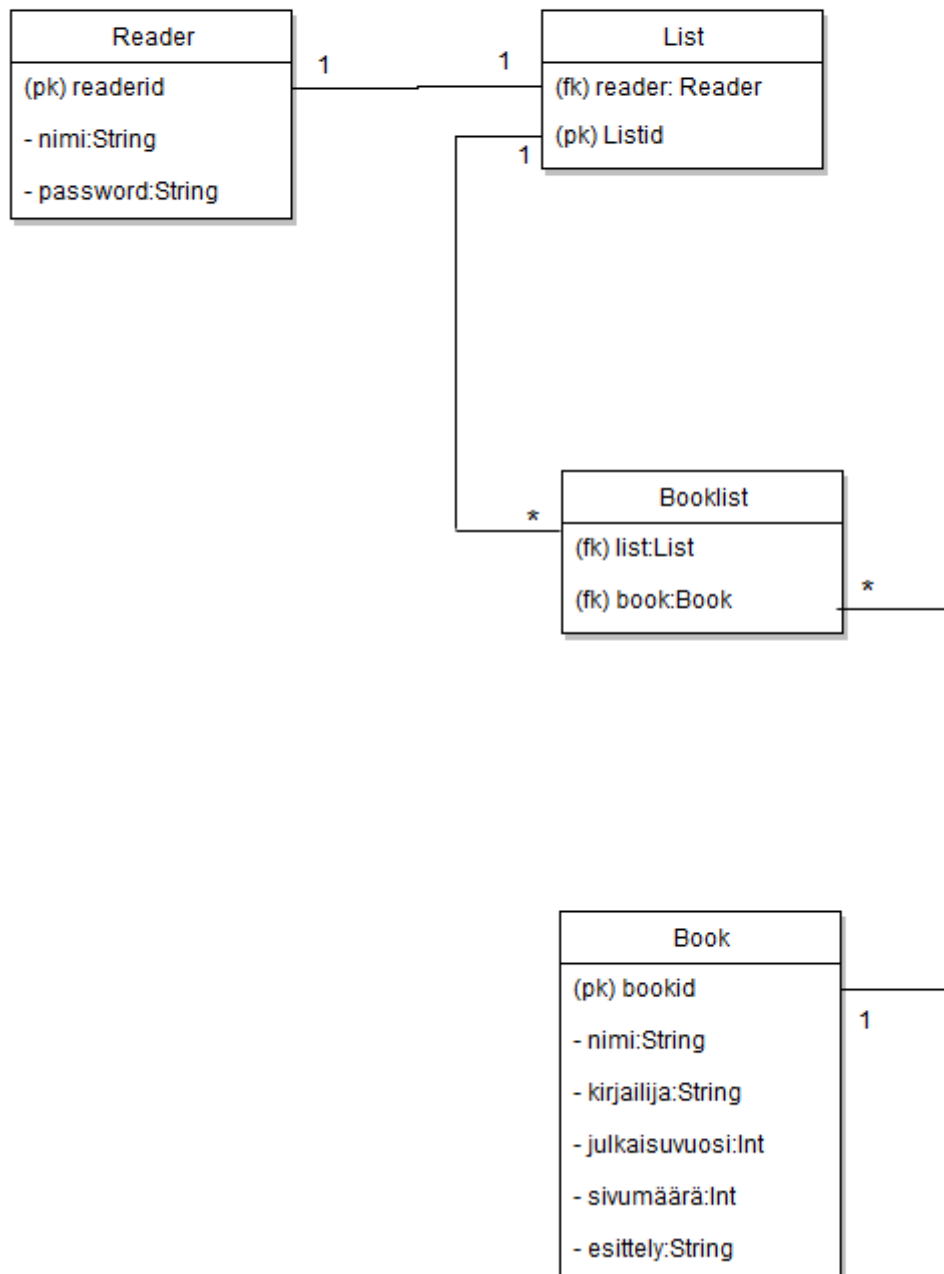
**Kuvaus:** "List" on jollekin tietylle käyttäjälle kuuluva henkilökohtainen kirjalista, joka sisältää käyttäjän lukemia kirjoja. Kirjalista sisältää nolla tai useita kirjoja.

### **Tietokohde: Book**

<b>Attribuutti:</b>	<b>Arvojoukko:</b>
Nimi	Merkkijono, max. 120 merkkiä
Kirjailija	Merkkijono, max. 80 merkkiä
1.painoksen julkaisuvuosi	Numeerinen
Sivumäärä	Numeerinen
Esittely	Merkkijono, max. 250 merkkiä

**Kuvaus:** Kaikki tietokannan kirjat ovat nähtävissä sovelluksen kaikille käyttäjille. Yksi kirja voi liittyä moniin eri kirjautuneiden käyttäjien kirjalistoihin (= List).

# Relaatiotietokantakaavio



## Järjestelmän yleisrakenne

Tietokantasovelluksessa noudatetaan MVC-mallia. Kontrollerit, mallit ja näkymät sijaitsevat hakemistoissa "controllers", "models" ja "views". Tiedosto "routes.php" hakemistossa "config" vastaanottaa tiettyihin polkuihin tulevat pyynnöt (post ja get), ja kutsuu oikean kontrollerin oikeaa metodia, joka muodostaa näkymän käyttäen hyväksi mallin tietoja.

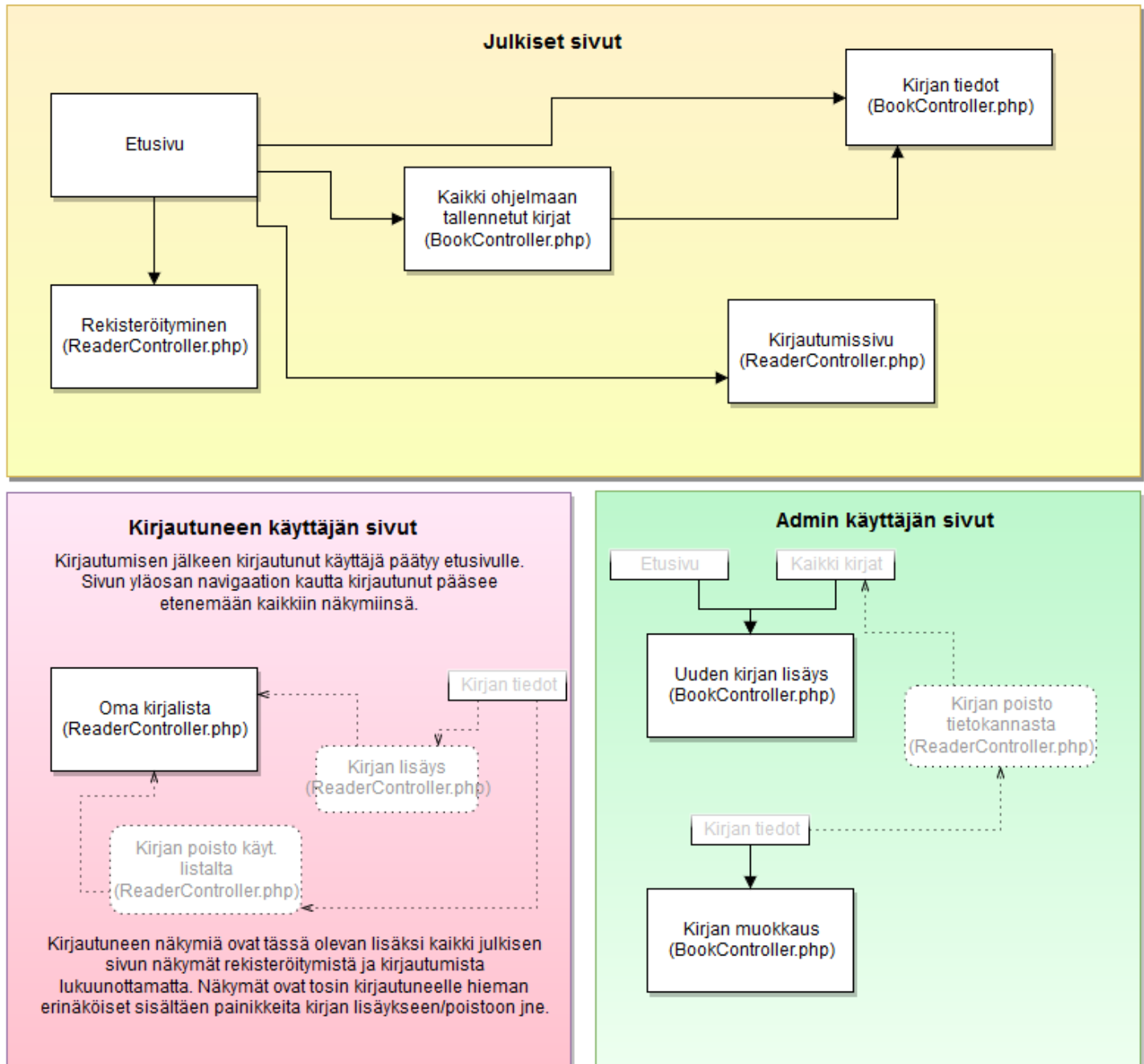
Sovelluksen kontrollereita ovat Book- ja ReaderController.php, joista BookController huolehtii kaikista näkymistä, joissa tarvitaan tietokannan jonkun kirjan tai useiden kirjojen tietoja. ReaderController huolehtii käyttäjään (= lukijaan) kuuluvista näkymistä, joissa osissa tarvitaan myös kirjojen tietoja. Ne kutsuvat malleja Reader.php ja Book.php, jotka huolehtivat tietokantaan kohdistuvien kyselyjen suorituksesta, sekä käyttäjän antamien syötteiden validoinnista yhdessä base\_model.php:n kanssa, jonka ne perivät.

Näkymät on jaettu kansioihin mallien nimien mukaan, ja ne kaikki perivät base.html-tiedoston, jossa määritellään esimerkiksi ylänavigaatiopalkki ja kutsutaan tyylitiedostoja. Bootstrapin lisäksi sivujen tyyliä määrittää assets/css hakemistossa oleva site.css.

Muut kansiot ja tiedostot ovat nimeltään suhteellisen itseään selittäviä. Kansio lib sisältää esimerkiksi base\_controller.php:n ja base\_model.php:n, jotka tarjoavat kontrollereille ja malleille toiminallisuuksia. Database.php:tä kutsutaan muodostettaessa tietokantayhteyttä, ja se kutsuu config-hakemiston database.php tiedostoa. View.php taas esimerkiksi luo pohjan näkymille make-metodissaan, jota kutsutaan kontrollerissa aina, kun halutaan näyttää näkymä käyttäjälle. View.php kutsuu twigia, sekä määrittää joitain kaikkien näkymien käyttöön saatavia muuttujia.

# Käyttöliittymä ja järjestelmän komponentit

## Käyttöliittymäkaavio





## Asennustiedot

Asenna sovellus kopioimalla sen tiedostot palvelimen nettiin näkyvään hakemistoon. Aseta sen jälkeen oikeat tiedot (= käytettävä tietokanta, käyttäjätunnus) hakemiston ”config” tiedostoihin environment.sh ja database.php.

## Käynnistys- / käyttöohje

Ohjelman etusivulle pääsee URLista:

<http://mclantta.users.cs.helsinki.fi/tsohaa/>

Sisään voi kirjautua testikäyttäjällä:

Käyttäjä: Testi

Salasana: Testinen

Jos haluaa lisätä / poistaa / muokata tietokannan kirjoja, pitää sisään kirjautua Admin -käyttäjällä:

Käyttäjä: Admin

Salasana: Admin