

# Kauppalista – dokumentaatio

Johannes Elmnäinen

014022100

[johannes.elmnainen@helsinki.fi](mailto:johannes.elmnainen@helsinki.fi)

# Sisältö

1. Johdanto.....	3
1.1. Järjestelmän tarkoitus.....	3
1.2. Järjestelmän kuvaus.....	3
1.3. Toteutusympäristö.....	3
1.4. Riippuvuudet.....	3
1.4.1. Palvelin.....	3
1.4.2. Käyttäjän selain.....	3
1.4.3. Tietokanta.....	3
2. Käyttötapaukset.....	3
2.1. Käyttäjäryhmät.....	3
2.1.1. Anonyymi.....	3
2.1.2. Kauppalainen.....	4
2.1.3. Listamestari.....	4
2.1.4. Listakollabori.....	4
2.2. Käyttötapauskaavio.....	5
3. Järjestelmän tietosisältö.....	6
4. Relaatiotietotietokantakaavio.....	6
5. Järjestelmän yleisrakenne.....	6
6. Käyttöliittymä ja järjestelmän komponentit.....	8
7. Asennus.....	9
8. Käynnistys- ja käyttöohje.....	10
9. Testaus, tunnetut bugit ja puutteet sekä jatkokehitysideat.....	10
9.1. Testaus.....	10
9.2. Puutteet.....	10
9.3. Jatkokehitysideat.....	10
10. Omat kokemukset.....	11

# 1. Johdanto

## 1.1. Järjestelmän tarkoitus

Järjestelmän tarkoitus on auttaa yksityishenkilöiden kauppamatkojen suunnittelua ja toteutusta.

## 1.2. Järjestelmän kuvaus

Kauppalista on tietokantasovellus, joka toimii virtuaalisena muistilistana kauppaostoksista. Yksineläjille sen päätoimintona on nimenomaan toimia mukaan otettavana listana ostoksista. Useamman hengen ruokakunnille ruokalista tarjoaa yhteistoimintamahdollisuuksia, kuten yhteistyössä koottavat ostoslistat, tarpeiden hankinnan hajauttamisen ja kulujen kirjanpidon.

## 1.3. Toteutusympäristö

Ohjelmointikielenä Kauppalistassa käytetään PHP:a. Tietokantana toimii MySQL-tietokanta. Sivuston ulkoasun toteutukseen käytetään CSS:ä SASS-esikäsittelijän kautta.

Työ löytyy osoitteesta <http://tsoha.elmnainen.fi>.

## 1.4. Riippuvuudet

### 1.4.1. Palvelin

Palvelimen tulee tukea PHP:a sekä sen PDO-kirjastoa.

### 1.4.2. Käyttäjän selain

Käyttäjän selainen tulee lukea standardia HTML5:ä sekä CSS:ää.

### 1.4.3. Tietokanta

Järjestelmä olettaa käyttävänsä MySQL-tietokantaa.

# 2. Käyttötapaukset

## 2.1. Käyttäjäryhmät

### 2.1.1. Anonyymi

Anonyymi on käyttäjä, joka ei ole kirjautunut järjestelmään

### **2.1.2. Kauppalainen**

Kauppalainen on rekisteröitynyt ja sisäänkirjautunut käyttäjä. Myös Listamestari ja Listakollabori kuuluvat tähän ryhmään.

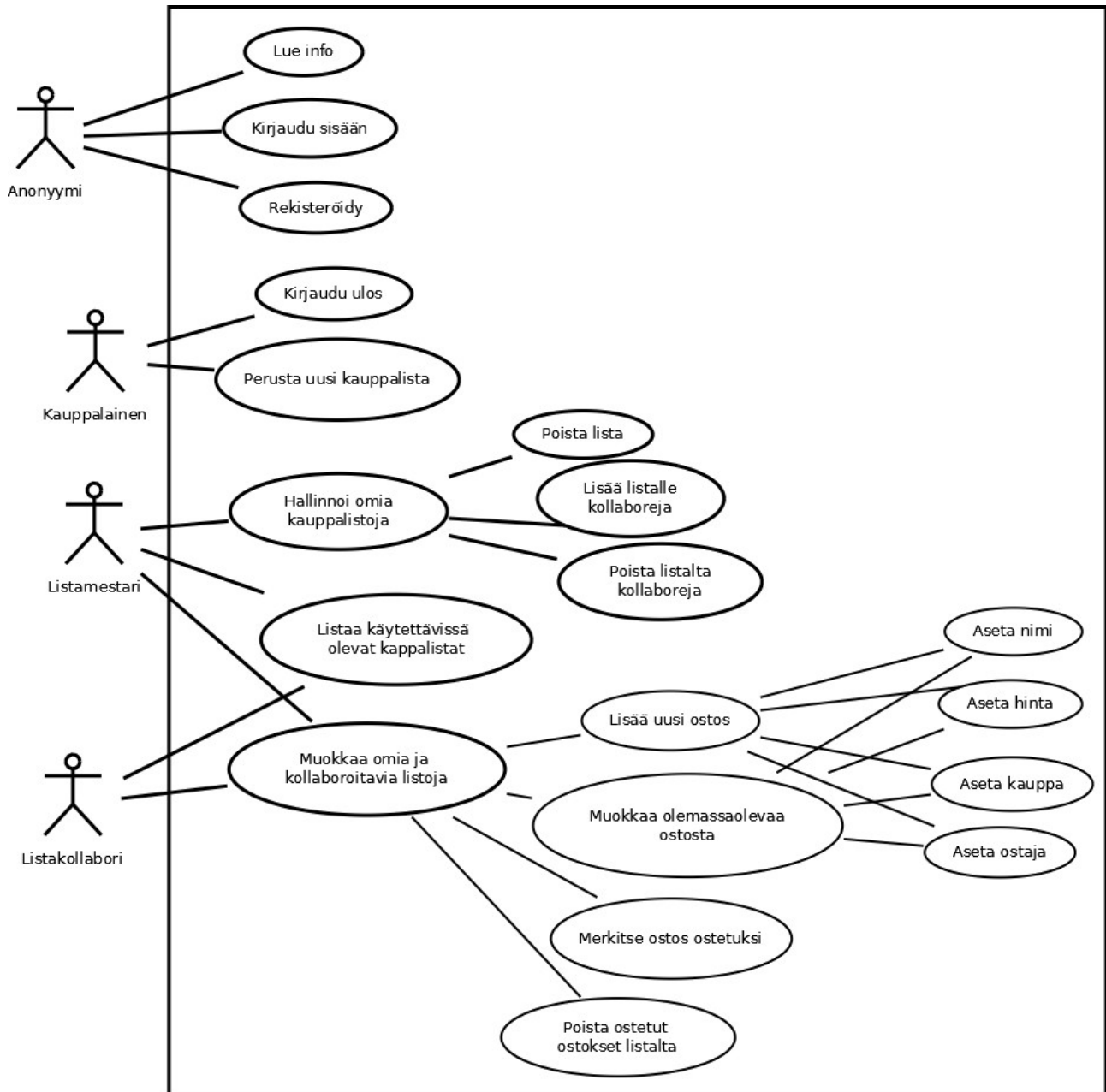
### **2.1.3. Listamestari**

Listamestari on Kauppalainen, joka on perustanut oman Kauppalistaan.

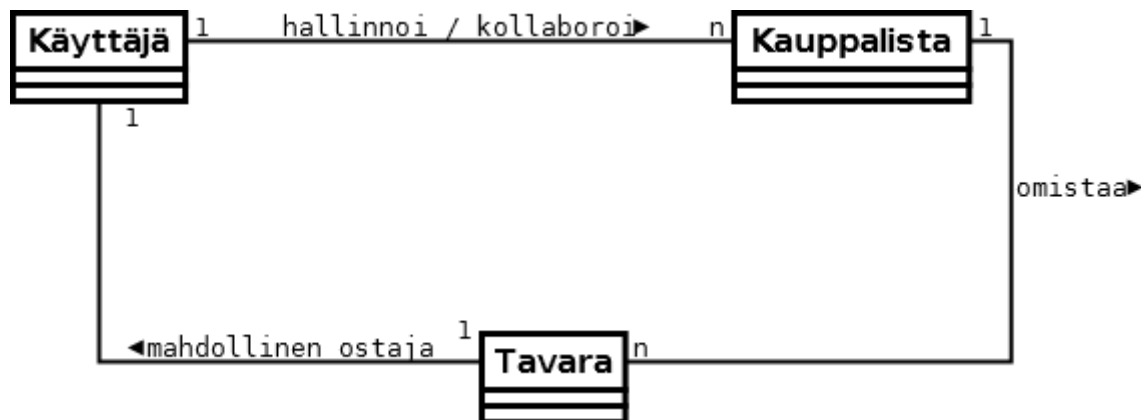
### **2.1.4. Listakollabori**

Listakollabori on Kauppalainen, jolle on myönnetty yhteistyöoikeus jonkin Listamestarin Kauppalistaan.

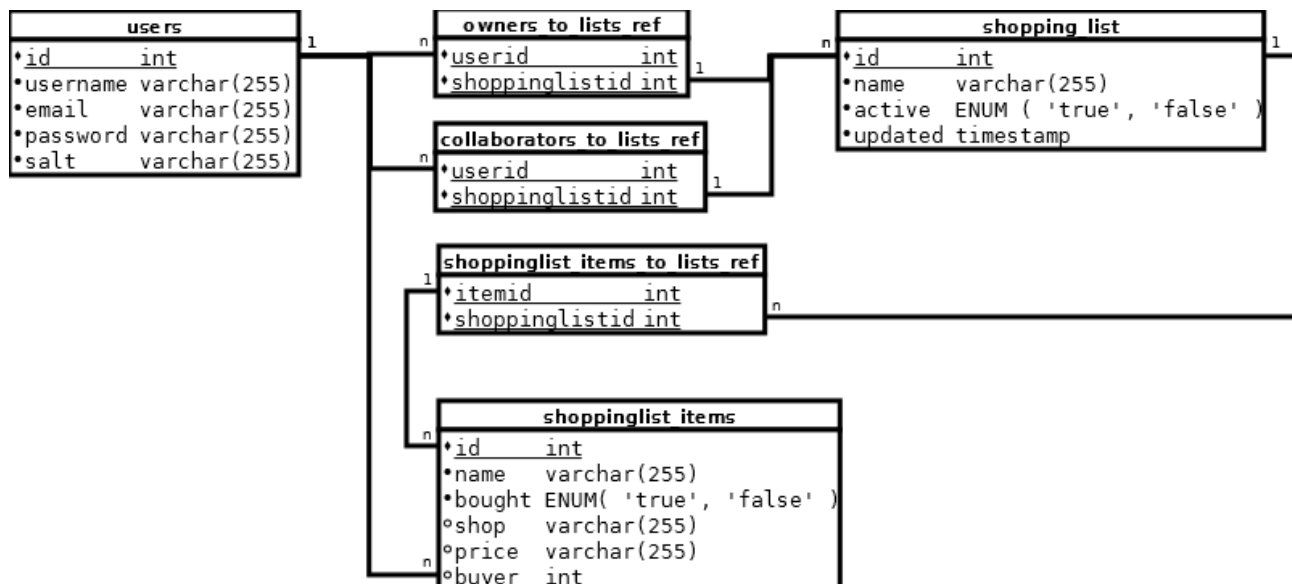
## 2.2. Käyttötapauskaavio



### 3. Järjestelmän tietosisältö



### 4. Relaatiotietotietokantakaavio



### 5. Järjestelmän yleisrakenne

Järjestelmän ytimenä pyörii core.php:stä löytyvä Core, joka on vastuussa tietokannan pystytyksestä, selaimen pyynnön parsimisesta ja niiden delegoinnista oikealle kontrollerille.

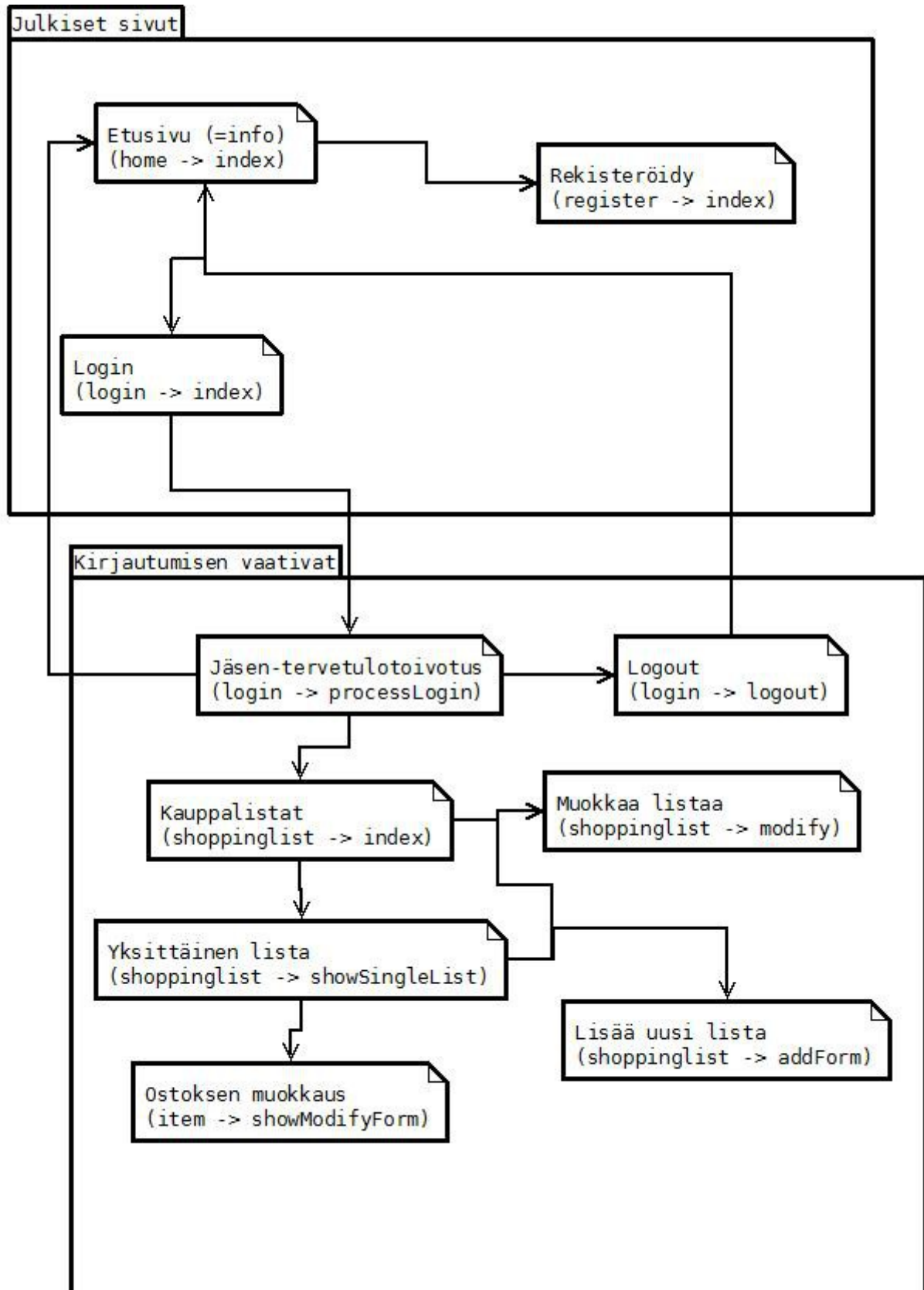
Tietokannan toiminnallisuus löytyy lib-kansion db-luokasta.

Asetustiedostot on jaettu kahteen juuresta löytyvän config.php:n sekä config-hakemistosta löytyvän dbconfigin välillä. Näin siksi, että dbconfig sisältää käyttäjätunnuksia, joten se on hyvä pitää .gitignoressa.

Kauppalistaa tehdessä on noudatettu MVC-mallia. Kontrollerit löytyvät hakemistosta controllers. Jokaisen kontrollerin käyttämät näkymät löytyvät hakemistosta views/\*kontrollerin\_nimi. Kontrollerit perivät perustoimintonsa controllerbase.php:lta. Kontrollerit käyttävät lisäksi mallin täyttämiseen palveluita, joiden tehtävänä on toimia työvälineinä tietokannan suuntaan, sekä domaineja, joiden tehtävänä on mallintaa yksittäisiä tietokannan oliota. Palvelut löytyvät services-hakemistosta ja domainit domains-hakemistossa. Domainit käyttävät domain-suffiksia ja palvelut service-suffiksia. Kaikki tiedostonimet on kirjoitettu pienellä.

Edellämainittujen lisäksi koodissa on vielä kirjoitushetkellä riippuvuuksia /models-hakemistoon, lähinnä login,- register- ja home-kontrollereiden osalta, mutta nämä tullaan refaktoroimaan aikanaan ulos ja näiden luokkien toteutukset siirtämään service-kansioon.

## 6. Käyttöliittymä ja järjestelmän komponentit



## 7. Asennus

Sovellus asennetaan kopioimalla sen tiedostot internetiin näkyvään tiedostoon. Tämän jälkeen tulee



ajaa kauppalista\_creation.sql tietokantaan sekä luoda config/dbconfig-tiedosto, joka asettaa \$dbconfig-muuttujaan taulukon, jonka osoittimina ovat "dns", "username" ja "password", ja joiden arvot viittaavat vastaavasti käytettävän tietokannan data source namea, käyttäjänimeä sekä salasanaa. Tämän jälkeen sovellus on valmis käytettäväksi.

## 8. Käynnistys- ja käyttöohje

Sovelluksen käyttöönotto tapahtuu menemällä sovellukselle osoitettuun osoitteeseen ja luomalla itselleen uusi käyttäjätunnus "rekisteröidy"-valikosta. Tämän jälkeen käyttäjä voi hallinoida kauppalistojaan menemällä valitsemalla "kauppalistat". Täällä käyttäjä voi lisätä tai poistaa listoja valitsemalla asianmukaiset linkit, lisätä listalle uusia ostoksia tai muuttaa listalla olevien ostosten statusta (punainen ruksi poistaa ostoksen, kynä merkitsee muokkausta ja vihreä siirtää ostoksen ostetuksi). Lisäksi käyttäjä voi hallinnoida omien listojensa kollaboreja.

## 9. Testaus, tunnetut bugit ja puutteet sekä jatkokehitysideat

### 9.1. Testaus

Sovellusta on testattu käyttötestauksella koko kehityksen ajan, eikä päätoiminnoissa pitäisi löytyä varsinaisia bugeja.

### 9.2. Puutteet

Sovellus ei tarkista, onko funktion kutsujalla oikeutta suorittaa kyseistä funktiota parametreina annetuilla arvoilla. Näin käyttäjäksi rekisteröityneellä hyökkääjällä on mahdollisuus esimerkiksi poistaa listalta kollabori yksinkertaisesti menemällä osoitteeseen `/?page=shoppinglist&action=removeCollaboratorByIDFromListByID&params=<käyttäjä_id>;<lista_id>`. Sovellus ei myöskään ole varautunut DDoS-tyyppisiin hyökkäyksiin vastaan. Yhdistettynä edelliseen tämä tarkoittaa, että hyökkääjä pystyy teoriassa tuhoamaan kaikki järjestelmän tiedot yksinkertaisesti pommittamalla tarpeellisen määrän `<käyttäjä_id>;<lista_id>`-yhdistelmäpyyntöjä.

### 9.3. Jatkokehitysideat

Tärkein jatkokehitysidea on hasRightToModify-funktio service-tasolle, joka hoitaisi kohdassa 9.2. kuvatus ongelman käsittelyn. Tämän jälkeen kannattaisi hoitaa yhden IP:n DoS-hyökkäysten esto. Seuraava kehitysidea voisi olla template-moottorin pulttaaminen järjestelmään sekä AJAX:n hyödyntäminen eri muokkausfunktioissa.

Koska pääkäyttö tulisi olemaan mobiilina, tulisi sovellukseen lisätä jonkinlainen "muista minut"-mahdollisuus mahdollisimman virtaviivaista mobiilikäyttöä varten sekä tarkistella käyttöliittymää niin, että listan ostosten manipulointi olisi mahdollisimman virtaviivaista mobiililaitteella. Yksi

tällainen idea voisi olla ”päälistan” merkintä. Päälista avattaisiin mobiilinäkymässä suoraan ainoaksi käsiteltäväksi listaksi. Muita virtaviivaistuksia voisivat olla esimerkiksi käyttöliittymäikonien määrän lisääminen ja koon kasvattaminen. Päälista voisi olla joko käyttäjän määrittelemä tai se, mihin käyttäjä on viimeksi lisännyt ostettavia tavaroita.

Näiden jälkeen sovellukseen voisi lisätä toiminnallisuutta niin, että se pitäisi kirjaa listan jokaisen käyttäjän kustannuksista ja antaisi tasaushinnat.

## **10. Omat kokemukset**

Tietokantasovelluksen teon vaikein osuus oli MVC:n pystyttäminen. PDO:n, php:n ja MySQL:n yhteistyöstä minulla oli jo valmiiksi kokemusta. Tämän kurssin suurimpana antina itselleni koenkin siis MVC:n ymmärtämisen jollain tasolla, sekä tutustumisen bootstrappiin.