

Aapo Korvenranta

Aineopintojen harjoitustyö: Tietokantasovellus
Syksy 2016, Periodi 1

Dokumentaatio

1 Johdanto

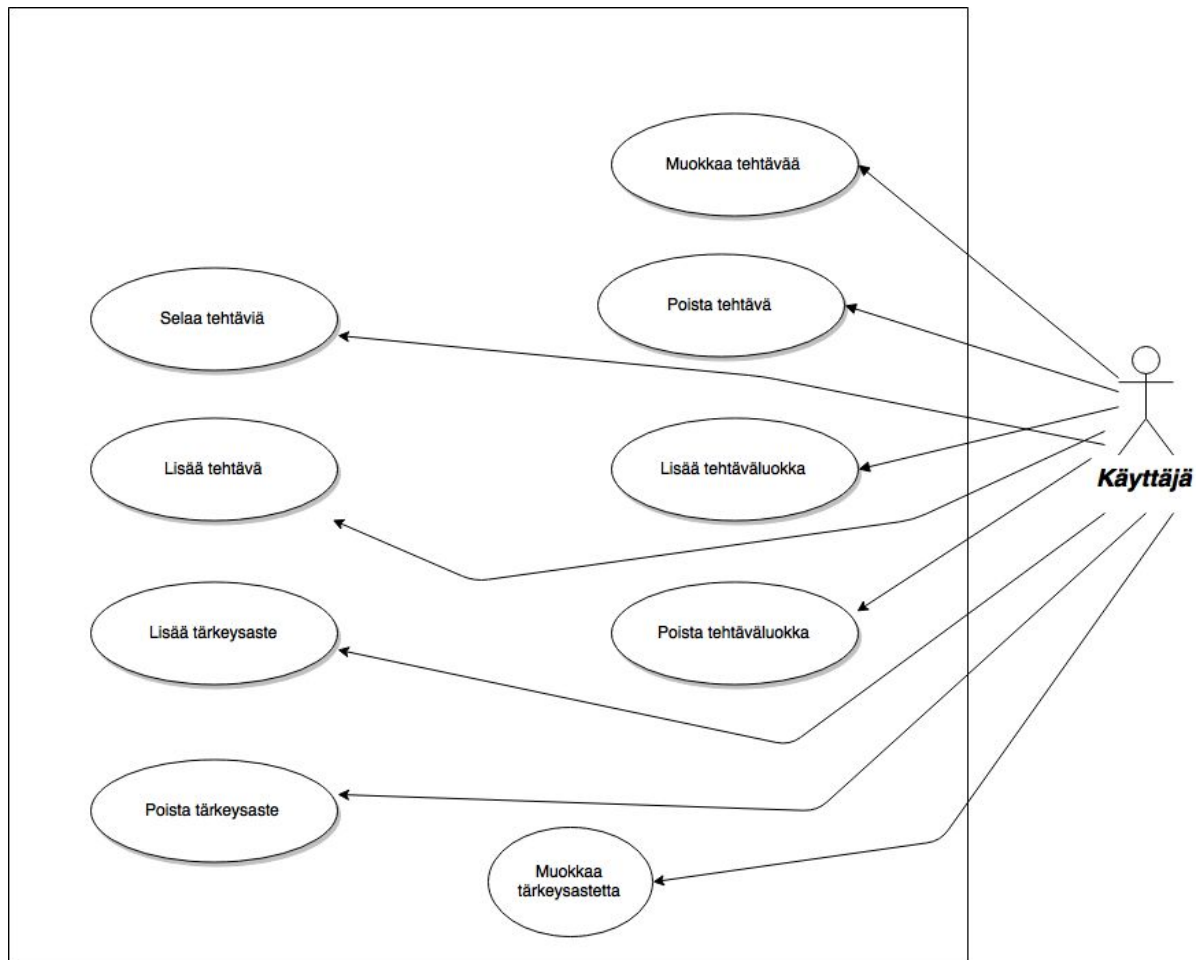
Muistilista- sovelluksen tarkoitus on kirjata ylös ja tarkastella tehtäviä jotka pitää suorittaa. Tehtäville annetaan tärkeysjärjestys, jonka mukaan ne lajitellaan. Tehtäviä voi myös luokitella erilaisiin luokkiin siten, että jokainen tehtävä voi kuulua useampaan luokkaan. Järjestelmään toteutetaan käyttäjänhallinta ja kirjautuminen, joten se voi olla useampien henkilöiden käytössä siten että kaikilla käyttäjillä on omat tehtävänsä ja tehtäväluokkansa.

Järjestelmä toteutetaan laitoksen users-palvelimella PHP- kielellä, joten palvelimella tulee olla PHP- tuki. Tietokantana käytetään PostgreSQL- tietokantaa. Selaimelle ei tule erityisiä vaatimuksia.

2 Käyttötapaukset

Sovelluksessa on vain yksi käyttäjäryhmä:

Käyttäjä on sovellukseen rekisteröitynyt ja kirjautunut käyttäjä, joka hallinnoi omia tehtäväluokkia, tärkeysjärjestyksiään ja tehtäviään.



Käyttötapausten kuvaukset:

Lisää tärkeysaste:

Käyttäjä lisää tärkeysasteen prioriteettijärjestysnumeron ja kuvauksen. Lisäys ei onnistu, jos järjestysnumero on jo olemassa.

Muokkaa tärkeysastetta:

Käyttäjä hakee tärkeysasteen ja voi muokata sen nimeä ja järjestysnumeroa.

Poista tärkeysaste:

Käyttäjä poistaa tärkeysasteen. Poisto ei onnistu jos tärkeysastetta on käytetty tehtävässä.

Lisää tehtäviäluokka:

Käyttäjä lisää tehtäviäluokan ja sen kuvauksen. Lisäys ei onnistu, jos luokka on jo olemassa.

Poista tehtäviäluokka:

Käyttäjä poistaa tehtäviäluokan. Poisto ei onnistu jos tehtäviäluokkaa on käytetty tehtävässä.

Lisää tehtävä:

Käyttäjä lisää tehtävän otsikon ja kuvauksen sekä määrittää tehtävälle tärkeysasteen ja yhden tai useampia tehtäväluokkia.

Muokkaa tehtävää:

Käyttäjä muuttaa tehtävän tietoja. Hän voi muuttaa tehtävän otsikkoa, kuvausta, tärkeysastetta ja tehtäväluokittelua.

Poista tehtävä:

Käyttäjä poistaa tehtävän saatuaan sen valmiiksi.

Selaa tehtäviä:

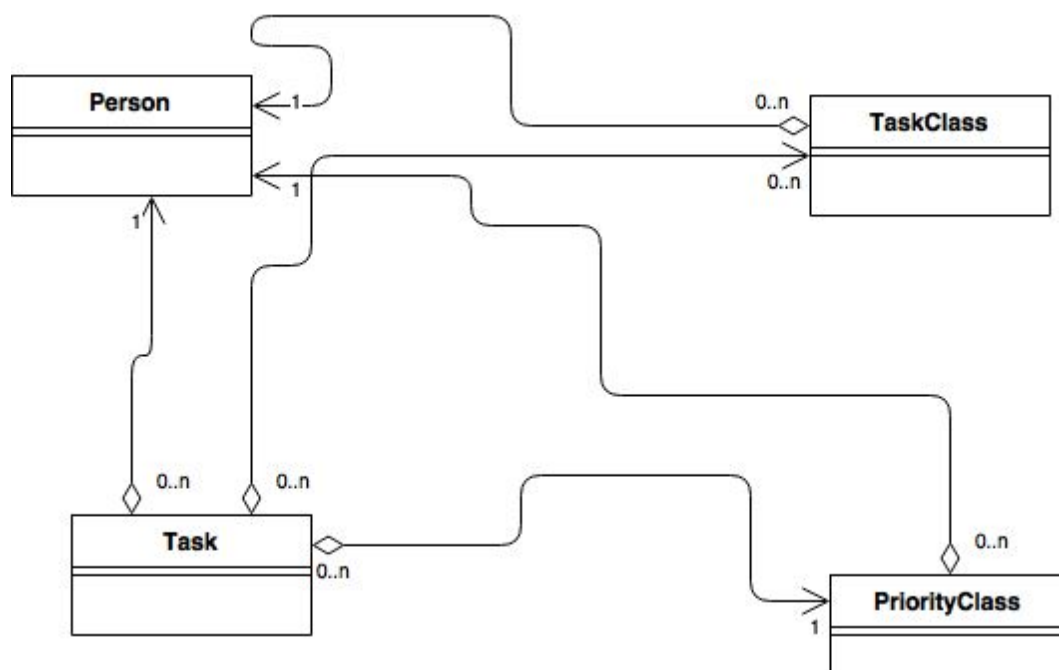
Käyttäjä hakee näytölle tehtävälistan. Tehtävälistalla näytetään tehtävän otsikko, tehtäväluokka ja tärkeysaste. Tehtävät on lajiteltu tärkeysasteen perusteella.

Listasta päästään yksittäisen tehtävän tietoihin, jossa näytetään em. tietojen lisäksi tehtävän tarkempi kuvaus.

Muut käyttötapaukset: rekisteröityminen, kirjautuminen.

Rekisteröitymistä ei toteuteta tässä vaiheessa, vaan pääkäyttäjä lisää käyttäjätunnukset suoraan tietokantaan.

3 Järjestelmän tietosisältö



Tietokohde Task:

Attribuutti	Arvojoukko	Kuvaus
TaskName	Merkkijono, max 50 merkkiä	Tehtävän otsikko
TaskDescription	Merkkijono, max 1000 merkkiä	Tehtävän kuvaus

Yksi tehtävä (task) voi kuulua ainoastaan yhdelle henkilölle. Samoin se voi kuulua ainoastaan yhteen tärkeysluokkaan. Tehtävällä voi kuitenkin olla useampia tehtäväluokkia.

Tietokohde Person:

Attribuutti	Arvojoukko	Kuvaus
FirstName	Merkkijono, max 20 merkkiä	Käyttäjän etunimi
LastName	Merkkijono, max 50 merkkiä	Käyttäjän sukunimi
UserId	Merkkijono, max 20 merkkiä	Käyttäjän käyttäjätunnus
Password	Merkkijono, max 20 merkkiä	Käyttäjän salasana

Henkilöllä voi olla useita tehtäviä, tehtäväluokkia sekä tärkeysasteita.

Tietokohde: TaskClass

Attribuutti	Arvojoukko	Kuvaus
TaskClassName	Merkkijono, max 50 merkkiä	Tehtäväluokan otsikko
TaskClassDescription	Merkkijono, max 1000 merkkiä	Tehtäväluokan kuvaus

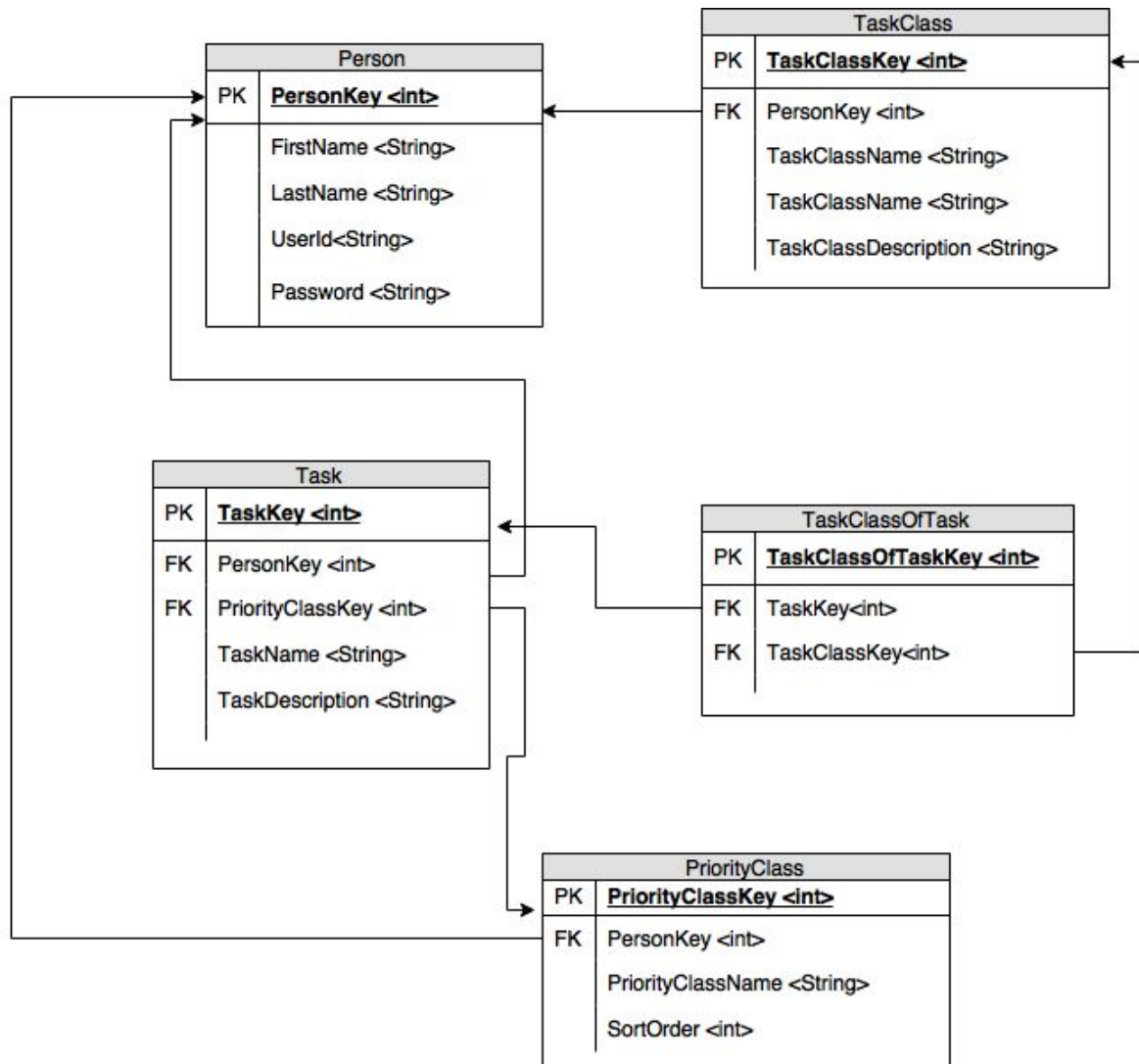
Tehtäväluokka (TaskClass) kuuluu aina yhdelle henkilölle.

Tietokohde: PriorityClass

Attribuutti	Arvojoukko	Kuvaus
PriorityClassName	Merkkijono, max 50 merkkiä	Tärkeysluokan nimi
SortOrder	Kokonaisluku	Tehtävälisan lajittelussa käytettävä järjestysnumero

Tärkeysluokka (PriorityClass) kuuluu aina yhdelle henkilölle.

4 Relaatiotietokantakaavio



5 Järjestelmän yleisrakenne

Muistilistasovellusta tehdessä on noudatettu MVC-mallia. Kontrollerit, näkymät ja mallit sijaitsevat hakemistoissa controllers, views- ja models. Views- hakemisto, joka sisältää käyttäjälle näytettävät html- sivut, on jaettu alihakemistoihin kokonaisuuksien mukaan:

- tasks: tehtävien tarkasteluun käsittelyyn tarkoitetut sivut
- priorityclasses: tärkeysluokkien käsittelyyn tarkoitetut sivut
- taskclasses: tehtävaluokkien käsittelyyn tarkoitetut sivut

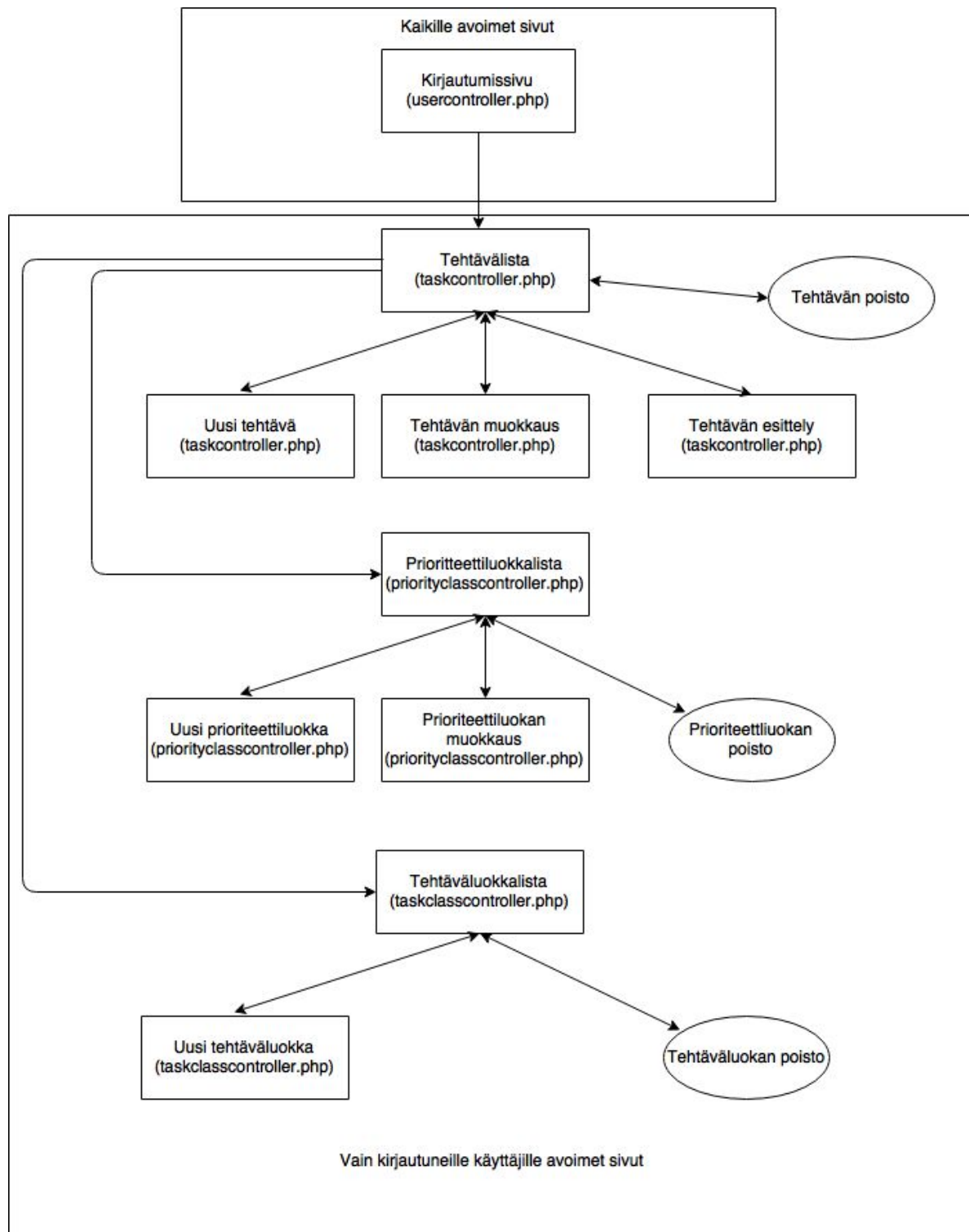
Omat apukirjastot, esimerkiksi malliluokkien yhteiset metodit sisältävä base_model.php on sijoitettu hakemistoon lib. ja kolmansien osapuolten kirjastot hakemistoon Vendor. Config-

hakemistossa on järjestelmän sovelluksen ympäristöön liittyvät luokat sovelluksen asennusta ja tietokantayhteyden muodostamista varten.

PhpDocumentorilla muodostettu Api- dokumentaatio on GitHubissa hakemistossa /doc/api.

Sisäänkirjautumisen yhteydessä sovellus tallentaa istuntoon käyttäjätaulun avaimen. Avainta käytetään suodattimena listojen haussa. Se myös tallennetaan kaikkiin tauluihin lisäys- ja muokkausoperaatioiden yhteydessä.

6 Käyttöliittymä ja järjestelmän komponentit



Lisäksi sivun yläreunassa on navigaatiopalkki, josta kirjautunut käyttäjä pääsee Tehtävälistaan, Prioriteettiluokkalistaan tai Tehtävuokkalistaan sekä kirjautumaan ulos.

7 Käynnistys- ja käyttöohje

Sovellus käynnistetään osoitteessa:

<https://korveaap.users.cs.helsinki.fi/Muistilista/>

Vain kirjautuneet käyttäjät voivat käyttää sovellusta. Kaikille kirjautuneille käyttäjille on käytössä samat toiminnallisuudet. Muille näytetään ainoastaan kirjautumissivu.

Sovelluksessa on kaksi testikäyttäjää (käyttäjätunnus ja salasana ovat case-sensitive):

Teppo Testaaja, käyttäjätunnus **teppo**, salasana **tt**

Kati Käyttäjä, käyttäjätunnus **Kati**, salasana **kk**

8 Asennusohje

Luo ensin tietokantataulut ajamalla tietokannassa skripti sql- hakemistossa oleva skripti create_tables.sql.

Tämän jälkeen asenna sovellus kopioimalla sen tiedostot palvelimen nettiin näkyvään hakemistoon (esim. usersin htdocs-hakemisto).

Asennuksen voi suorittaa myös päivittämällä kohdehakemisto ja käyttäjätunnus hakemistoon config/environment.sh ja ajamalla tämän jälkeen create_tables.sh ja deploy.sh- skriptit.

Asennuksen jälkeen aseta tietokannan yhteystiedot oikeaksi tiedostoon config/database.php.

9 Testaus & Jatkokehitys

Olen testannut sovelluksen kaikki toiminnallisuudet. Annoin myös toiselle testaajalle tehtäväksi tehtäväksi rikkoa sovellus (=aiheuttaa hallitsematon virhetilanne). Bugeja löytyi lähinnä tietojen validointiin liittyen, mutta kaikki löytyneet bugit on korjattua. Emme osanneet testata mahdollisia tapoja kiertää käyttäjän autentikointi, joten todennäköisesti näppärä hakkeri sen pystyisi tekemään.

Jatkokehitykseksi jäi rekisteröintitoiminnallisuuden ja listojen suodatustoiminnallisuuksien toteuttaminen. Sovelluksen ulkoasua voisi myös parantaa.

10 Kokemukset & Opit

Helppoa oli itse tietokantaan liittyvät hommat, eli kannan suunnittelu, toteutus ja sql- lauseiden kirjoittaminen.

Vaikeaa oli aluksi MVC- mallin hahmottaminen, mutta se selkeni loppua kohden ollen näin suurin oppi tästä projektista.

Javaan tottuneena Php:n syntaksi aiheutti myös jonkun verran haasteita, mutta hyvillä esimerkeillä niistä selvisi kunnialla.