

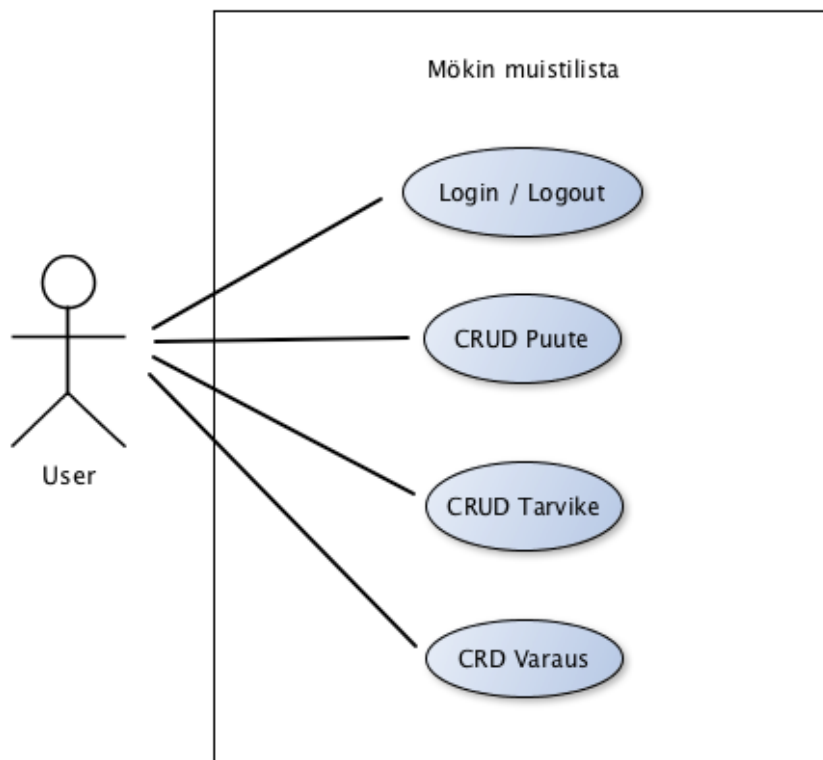
Mökin muistilista

Järjestelmän tarkoituksena on toimia työkaluna mökin puutteiden ja tarpeiden kirjaukseen. Sen lisäksi käyttäjät voivat päivittää tietoja asioista, joita mökiltä jo löytyy. Käyttäjät voivat myös merkitä mökin tietoihin ajankohtia, jolloin he olisivat menossa mökille. Usea käyttäjä voi varata itselleen mökkivierailun samaan aikaan, päällekkäyksiä ei siis rajoiteta.

Järjestelmä pyrkii tarjoamaan ratkaisun mökin käyttäjille, jonka avulla he tietävät suunnilleen mitä edelliset käyttäjät ovat mökille jättäneet ja mitä sinne täytyy hankkia.

Järjestelmän tietokannaksi valitaan PostgreSQL ja ohjelmointikieleksi PHP. Sovellus asennetaan laitoksen Users -palvelimen Apache web-palvelimelle.

Käyttötapaukset



Userin käyttötapaukset

Mökin puutteiden tarkastelu:

Käyttäjä voi kirjaututtuaan tarkastella mitä mökille tarvitaan ja päivittää listaa.

Mökin tavaroiden tarkastelu:

Käyttäjä voi kirjaututtuaan tarkastella mitä mökiltä löytyy ja muokata listan sisältöä.

Mökin käytön tarkastelu:

Käyttäjä voi tarkastella onko mökille menossa muita samaan aikaan. Sen lisäksi hän voi luoda uusia vierailuja sekä peruuttaa vierailuja. Vierailuja ei kuitenkaan voi muokata.

Muita käyttötapauksia:

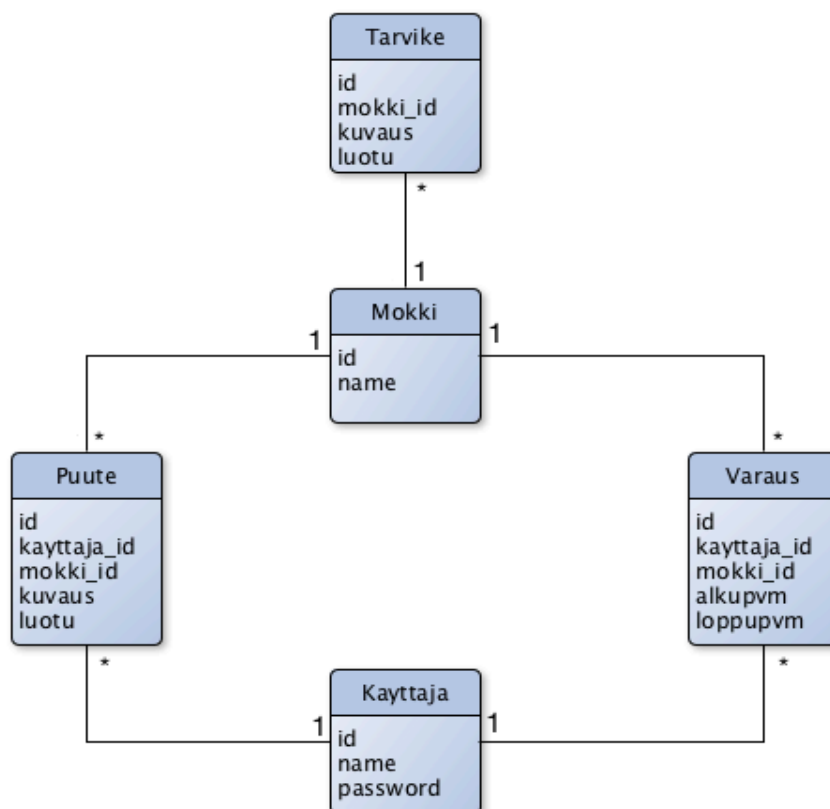
Login ja logout.

Käyttäjärühmät

User

User on sivuston tavallinen käyttäjä. Hän voi lisätä, katsoa, muokata ja poistaa mökin tarpeissa olevia asioita ja päivittää tietoja mitä mökillä on. Sen lisäksi käyttäjä voi luoda uusia vierailuja, sekä poistaa jo olemassaolevia vierailuja. Käyttäjä voi myös kirjautua sisään ja ulos. Käyttäjätunnuksia ei voi luoda järjestelmästä vaan ne täytyy luoda kantaan järjestelmän ylläpidon toimesta.

Järjestelmän tietosisältö



Tietokohde: Käyttäjä

Attribuutti	Arvojoukko	Kuvailu
Käyttäjätunnus	Merkkijono, max 15 merkkiä	Jokainen käyttäjä tarvitsee uniikin käyttäjätunnuksen palveluun

Salasana	Merkkijono, max 50 merkkiä	Jokaisella käyttäjällä on tietokantaan talletettu salasana kirjautumista varten
----------	----------------------------	---

Tietokohde: Mökki

Attribuutti	Arvojoukko	Kuvailu
Nimi	Merkkijono, max 30 merkkiä	Jokaisella mökillä on nimi

Tietokohde: Varaus

Attribuutti	Arvojoukko	Kuvailu
Alkupäivämäärä	Date	Vierailun alkamisajankohta
Loppupäivämäärä	Date	Vierailun loppumisajankohta

Varaus kuuluu aina yhdelle käyttäjälle, sekä liittyy yhteen mökkiin.

Tietokohde: Puute

Attribuutti	Arvojoukko	Kuvailu
Kuvaus	Merkkijono, max 200 merkkiä	Jokaiseen puutteeseen liittyy kuvaus, josta ilmenee mitä puuttuu
Ajankohta	Date	Jokaiseen puutteeseen tallennetaan ajankohta, jolloin se on luoto

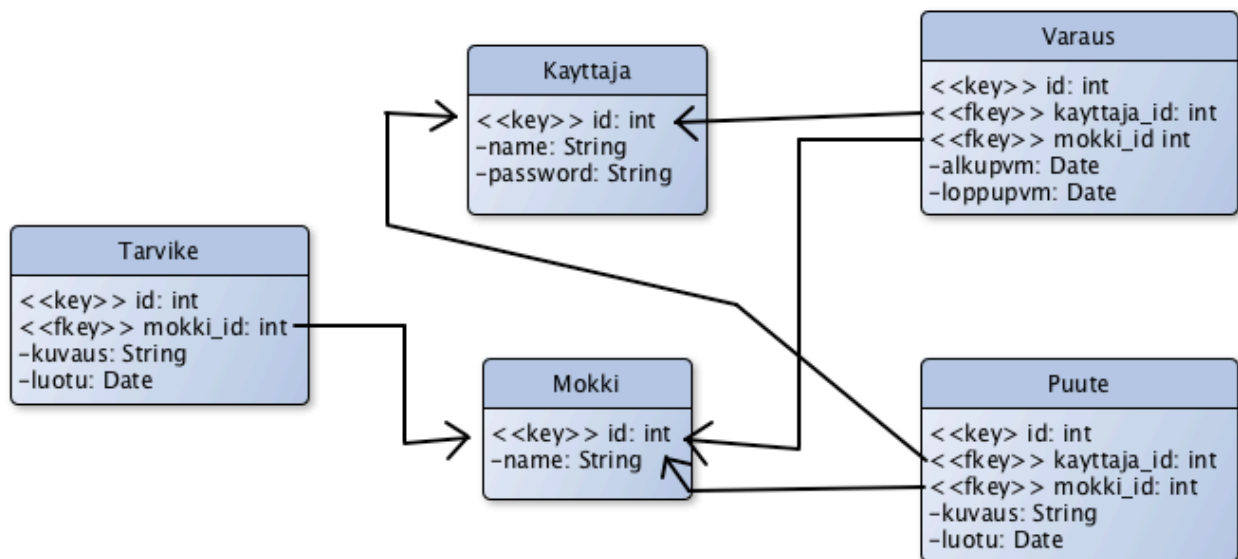
Puute kuuluu aina yhdelle käyttäjälle (käyttäjä, joka luo / muokkaa puutteen) ja kohdistuu yhteen mökkiin.

Tietokohde: Tarvike

Attribuutti	Arvojoukko	Kuvailu
Kuvaus	Merkkijono, max 200 merkkiä	Jokaiseen tarvikkeeseen liittyy kuvaus, joka kertoo mitä mökiltä löytyy
Ajankohta	Date	Jokaiseen tarvikkeeseen liittyy päivämää, jolloin tarvike on listattu olevaksi mökillä

Tarvike liittyy aina yhteen mökkiin. Yhdessä mökissä on luonnollisesti monia tarvikkeita.

Relaatiotietokantakaavio



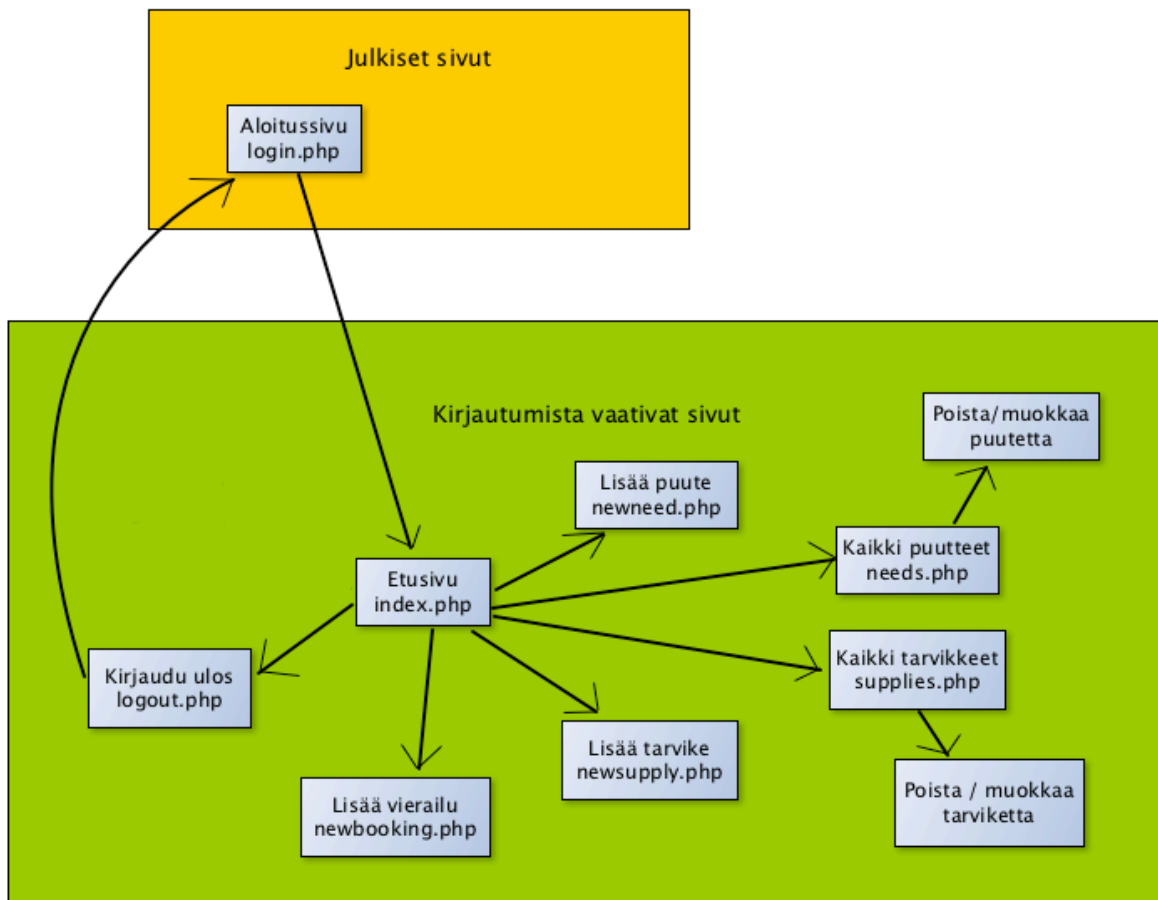
Järjestelmän yleirakenne

Olen sovelluksessani pyrkinyt noudattamaan MVC -mallia. Tiedostoja on kuitenkin melko suuri määrä ja kontrollereja on erittäin monta vain muutaman käyttötapauksen toteuttamiseksi.

Sovelluksen kaikki mallit sijaitsevat /libs/models/ -kansiossa. Jokaista tietokohdetta vastaa oma luokka, joka sisältää kaiken luokan tietokantayhteyksissä käytettävän koodin. /libs/ -kansiossa sijaitsee kaksi tärkeää tiedostoa: common.php sekä tietokantayhteys.php. tietokantayhteys.php:n avulla avataan yhteys kantaan. common.php:lla käynnistetään sessio sekä se sisältää apumetodin, jolla näytetään kontrollerien kutsumia näkymiä. Kontrollerit jäivät juurikansioon, sillä niiden siirtäminen aiheutti ongelmia polkumäärittelyjen kanssa ja näiden korjaukseen ei riittänyt aika.

/views/ -kansio sisältää ohjelman kaiken html:n ja näkymät, jotka muodostavat sivujen rungon. Kontrolleri antaa tarvittavan datan viewille. Sivun pohjana toimii pohja.php -tiedosto. css -tyylit sivuille ovat kansiossa /css/. Seuraavaksi esiteltävän käyttöliittymän sivut löytyvät sivuston hakemiston juuresta.

Käyttöliittymä



Asennustiedot

Mikäli haluat asentaa ohjelman esimerkiksi omalle palvelimellesi, hae Github -hakemiston sisältö kansioon, joka näkyy selaimessa ulospäin palvelimeltäsi. Tiedostojen polut on määritelty suhteellisesti, joten niitä ei tarvitse muokata.

Kansiossa `libs` on tiedosto `tietokantayhteys.php`, jonka avulla kantaan otetaan yhteys tarvittaessa. Tarkista, miten saat yhteyden omaan tietokantaasi. Jos se on `psql`, saman tiedoston pitäisi toimia yhteyden muodostamisessa. Tietokannan voit alustaa kansion `sql` -tiedostoilla.

Käyttöohje

Ohjelma osoitteessa `laxlax.users.cs.helsinki.fi`

Ohjelmaa käyttäksesi sinun tulee kirjautua sisään, tunnus: `testi`, salasana: `testaus` tai tunnus: `testaaja`, salasana: `testeri`.

Testaus, tunnetut bugit ja puutteet sekä jatkokehitysideat

Ohjelma on toteutettu erittäin suoraviivaisesti ja siinä ei ole oikeastaan mitään ylimääräistä. Tietokantahaut ovat erittäin yksinkertaisia ja talletettavat tiedot eivät ole määrältään suuria tai monimutkaisia. Jatkokehityksenä voisi toteuttaa esimerkiksi admin-käyttäjän, joka voisi luoda selaimen kautta uusia mökkejä sekä käyttäjätunnuksia. Sen lisäksi käyttäjälle voisi antaa listauksen vain hänen omista vierailuistaan. Samoin tarvikkeiden sekä puutteiden sivulle voisi luoda hakutoiminnon jonkin asian olemassaolon nopeampaa tarkistusta varten. Samoin käyttäjälle olisi varmasti ystävällistä, jos hän pystyisi suodattamaan näkyville vain tietyn mökin tietoja kerrallaan.

Testaus on toteutettu täysin selaimen kautta. Lomakkeita testattaessa on käytetty tyhmiä syötteitä sekä selaimen back-nappia. Tällä hetkellä tekstikentistä ei tarkisteta muuta kuin, että ovatko ne tyhjä ja tieto tallennetaan kantaan. En kuitenkaan onnistunut saamaan sivuja pahemmin rikki syöttämällä html:ää, php:ta tai javascriptia lomakkeisiin. Olen katsellut ulkoasua vain oman näyttoni kautta, joten on mahdollista, että IE:llä tai jollain muun kokoisella näytöllä leiska hieman tökkii.

Muutamia puutteita ja huomiota:

- Uuden vierailun luonnissa kentän tyyppi on html5 date, jota tukee vain chrome näyttämällä kalenterinäkymän.
- Vierailun voi luoda menneelle päivämäärälle mikä on hieman hölmöä mutta etusivulla näytetään vain vierailut, jotka eivät ole vielä päättyneet.
- Selaimen back-näppäin aiheuttaa paikoittain ongelmia jos sitä päättää painaa mutta tietokantaan se ei vaikuta. Esimerkiksi jos välissä tekee logoutin mutta sitten kirjautuu takaisin, pääsee backilla edellisen istunnon sivuja selaamaan taaksepäin.
- Sessio ei vanhene koskaan mikäli selain pysyy auki.
- Salasanat kannassa tekstinä

Omat kokemukset

Ensimmäinen viikko sujui melko mukavasti. Toinen sekä kolmas viikko aiheuttivat eniten ongelmia PHP:n kanssa, sillä se oli entuudestaan erittäin huonosti tuttu. Neljännellä viikolla työ alkoi jo sujua jouhevammin. Koodissa on kuitenkin todella paljon copy-pastea eri modelien ja kontrollerien välillä. Ehkä joku päivä PHP:ta voisi kokeilla jonkin frameworkin kanssa mutta kyllä Rails tämän hakkaa 100-0. Kurssin loppua kohden laatu myös hieman kärsi muiden kurssien painaessa päälle. Sain kuitenkin omasta mielestäni valmiiksi sen mitä tavoittelin ja myös bootstrap -osaaminen karttui jälleen hieman lisää!

Tietokannan käyttö tässä työssä on melko low-level koska työ ei vaadi ihmeempää hakuja tai liitoksia. Laitoksen users-palvelin oli erittäin positiivinen yllätys sen suhteen kuinka helppo PHP-PostgreSQL oli pystyttävä viikolla 1. Lopuksi vielä tajusin, että Puutteella sekä Tarvikkeella on erikseen tieto luontiajasta, vaikka tämän saisi suoraan kannasta, eikä sitä siis tarvitsisi tallentaa erikseen. Tämä työ oli mielenkiintoisempi kuin Javalabra ☺