

Tietokantasovellus Muistilista

Noora Imberg

kevät 2014

Sisällys

Sisällys

[1. Johdanto](#)

[2. Yleiskuva järjestelmästä](#)

[Käyttötapauskaavio](#)

[Käyttäjäryhmät](#)

[3. Käyttötapaukset](#)

[Käyttäjän käyttötapaukset](#)

[4. Järjestelmän tietosisältö](#)

[Tietokohdekuvaukset](#)

[5. Relaatiotietokantakaavio](#)

[6. Järjestelmän yleisrakenne](#)

[7. Järjestelmän komponentit](#)

[Kontrollerit](#)

[Näkymät views/](#)

[Tietokantataulut sql/](#)

[Yleiskäytöiset tiedostot libs/](#)

[Mallit libs/models/](#)

[JavaScript js/](#)

[css-tyylitiedostot css/](#)

[8. Käyttöliittymä](#)

[9. Asennustiedot](#)

[10. Käyttöohje](#)

[11. Testaus, puutteet ja jatkokehityssideat](#)

[12. Omat kokemukset](#)

[13. Liitteet](#)

1. Johdanto

Muistilista on tietokantasovellus-harjoitustyökurssin projektti, jonka tehtävänanto löytyy täältä: <http://advancedkittenry.github.io/suunnittelua/aiheet/Muistilista.html> .

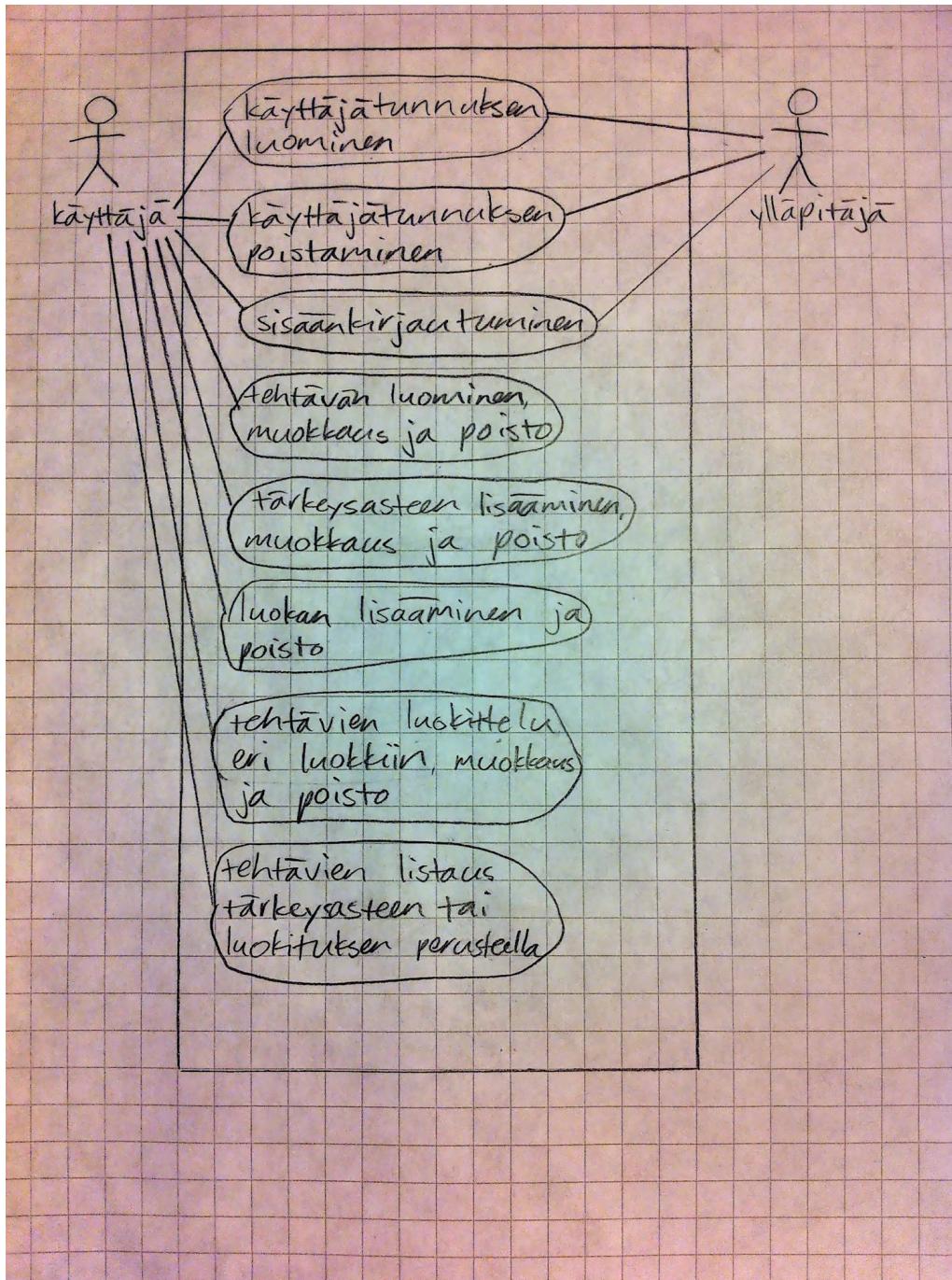
Muistilista on palvelu, johon pystyy sisäänkirjautumisen jälkeen lisäämään itselleen tehtäviä. Tehtävät voivat olla esimerkiksi asioita, jotka käyttäjän pitäisi muistaa tehdä. Muistilistaan lisättyjä tehtäviä on mahdollista muokata ja poistaa sekä antaa niille tärkeysasteita. Tehtävät voi lajitella tärkeysasteen perusteella tai niitä voi ryhmitellä erilaisiin luokkiin. Yksi tehtävä voi kuulua useampaan eri luokkaan.

Muistilistan avulla käyttäjä voi helposti ylläpitää listaa vielä tekemättömistä tehtävistä ja etsiä sieltä erilaisten kriteerien perusteella tehtäviä, jotka hänen pitäisi seuraavaksi tehdä. Kriteerinä voi olla tehtäville annettu tärkeysaste tai luokat, joihin ne kuuluvat.

Muistilista on toteutettu käytäen php:tä ja PostgreSQL-tietokantaa. Sovellus on laitoksen users-palvelimella.

2. Yleiskuva järjestelmästä

Käyttötapauskaavio



Käyttäjäryhmät

Käyttäjä on kuka tahansa, joka haluaa käyttää muistilistapalvelua omien tehtäviensä tallentamiseen.

3. Käyttötapaukset

Käyttäjän käyttötapaukset

Kun uusi käyttäjä haluaa aloittaa palvelun käyttämisen, hänen on luotava itselleen käyttäjätunnus. Myöhemmin palvelua käytäessään käyttäjän on ensin kirjauduttava sisään käyttäen omaa käyttäjätunnustaan ja salasanaansa. Sisäänkirjautumisen jälkeen käyttäjä voi tehdä seuraavia asioita:

Uuden tehtävän lisäys: Käyttäjä voi lisätä itselleen uuden tehtävän ja antaa sille tärkeysasteen. Jos käyttäjä on aiemmin luonut luokkia, joihin tehtäviä voi luokitella, hän voi myös lisätä uuden tehtävän näihin luokkiin.

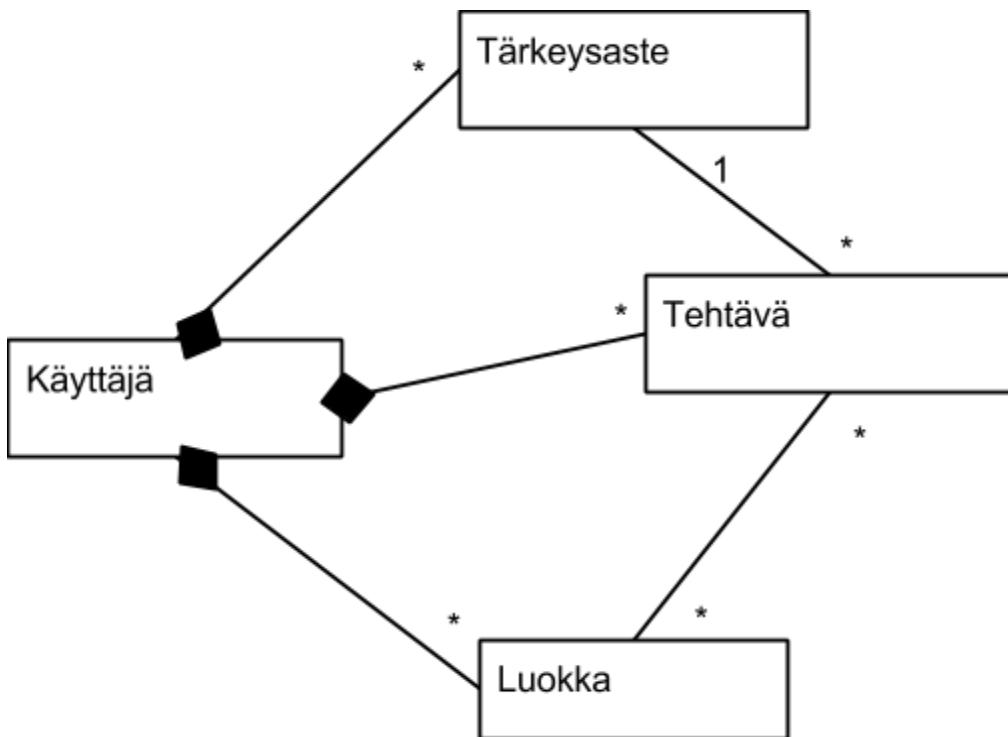
Tehtävän muokkaus ja poisto: Käyttäjälle näytetään lista tehtävistä. Hänen voi valita jonkin näistä ja muokata sen kuvausta, tärkeysastetta tai luokkia. Hänen voi myös poistaa koko tehtävän.

Luokkien lisäys ja muokkaus: Käyttäjälle näytetään lista olemassaolevista tehtäväluokista. Käyttäjä voi lisätä itselleen uuden tehtäväluokan, johon voi myöhemmin lisätä omia tehtäviään. Luokkia voi myös muokata ja poistaa.

Tärkeysasteiden lisäys ja muokkaus: Käyttäjälle näytetään lista olemassaolevista tärkeysasteista. Käyttäjä voi lisätä itselleen uuden tärkeysasteen, johon voi sitten luokitella tehtäviä. Tärkeysasteita voi myös muokata ja poistaa.

Tehtävien listaus: Käyttäjä voi listata tehtävänsä tiettyjen kriteerien perusteella. Hänen voi tarkastella esimerkiksi kaikkia tehtäviään, tietyn tärkeysasteen tehtäviä tai tiettyyn luokkaan kuuluvia tehtäviä. Käyttäjälle näytetään lista tehtävistä, jotka vastaavat annettuja kriteereitä.

4. Järjestelmän tietosisältö



Käyttäjällä voi olla useita tehtäviä, luokkia ja tärkeysasteita. Kaikki tehtävät, luokat ja tärkeysasteet kuuluvat yhdelle käyttäjälle. Tehtävällä on yksi tärkeysaste ja se voi kuulua yhteen tai useampaan luokkaan. Tärkeysasteseen ja luokkaan voi kuulua useampia tehtäviä.

Tietokohdekuvaukset

Tietokohde: Käyttäjä:

Attribuutti	Arvojoukko	Kuvailu
id	kokonaisluku (32-bit)	käyttäjän tunnistenumero
nimi	merkkijono, max 50 merkkiä	käyttäjän tervehtimiseen käytetty nimi
käyttäjätunnus	merkkijono, max 20 merkkiä	käyttäjän käyttäjätunnus, jota käytetään sisäänskirjautumiseen

salasana	merkkijono, max 20 merkkiä	käyttäjän salasana, jota käytetään sisäänsijautumiseen
----------	----------------------------	--

Käyttäjä on palvelun käyttäjä, jolla on nimi, käyttäjätunnus ja salasana. Nimeä käytetään käyttäjän tervehtimiseen. Tunnusta ja salasanaa käytetään sisäänsijautumiseen.

Tietokohde: Tehtävä:

Attribuutti	Arvojoukko	Kuvailu
id	kokonaisluku (32-bit)	tehtävän tunnistenumero
kuvaus	merkkijono, max 1000 merkkiä	tehtävän kuvailu
käyttäjä_id	kokonaisluku (32-bit)	kertoo sen käyttäjän tehtävästä on kyse
tärkeysaste_id	kokonaisluku (32-bit)	kertoo tehtävään liittyvän tärkeysasteen

Tehtävät liittyvät aina johonkin käyttäjään. Tehtävillä on kuvaus, jossa kerrotaan mitä käyttäjän pitäisi muistaa tehdä. Tehtävällä voi olla yksi tärkeysaste.

Tietokohde: Tärkeysaste:

Attribuutti	Arvojoukko	Kuvailu
id	kokonaisluku (32-bit)	tärkeysasteen tunnistenumero
nimi	merkkijono, max 255 merkkiä	tärkeysasteen nimi
käyttäjä_id	kokonaisluku (32-bit)	kertoo sen käyttäjän tärkeysasteesta on kyse

Tärkeysaste liittyy aina johonkin käyttäjään. Tärkeysasteella on nimi, joka kertoo joitain tehtävien tärkeydestä.

Tietokohde: Luokka:

Attribuutti	Arvojoukko	Kuvailu
id	kokonaisluku (32-bit)	luokan tunnistenumero
nimi	merkkijono, max 255 merkkiä	luokan nimi
käyttäjä_id	kokonaisluku (32-bit)	mille käyttäjälle luokka kuuluu

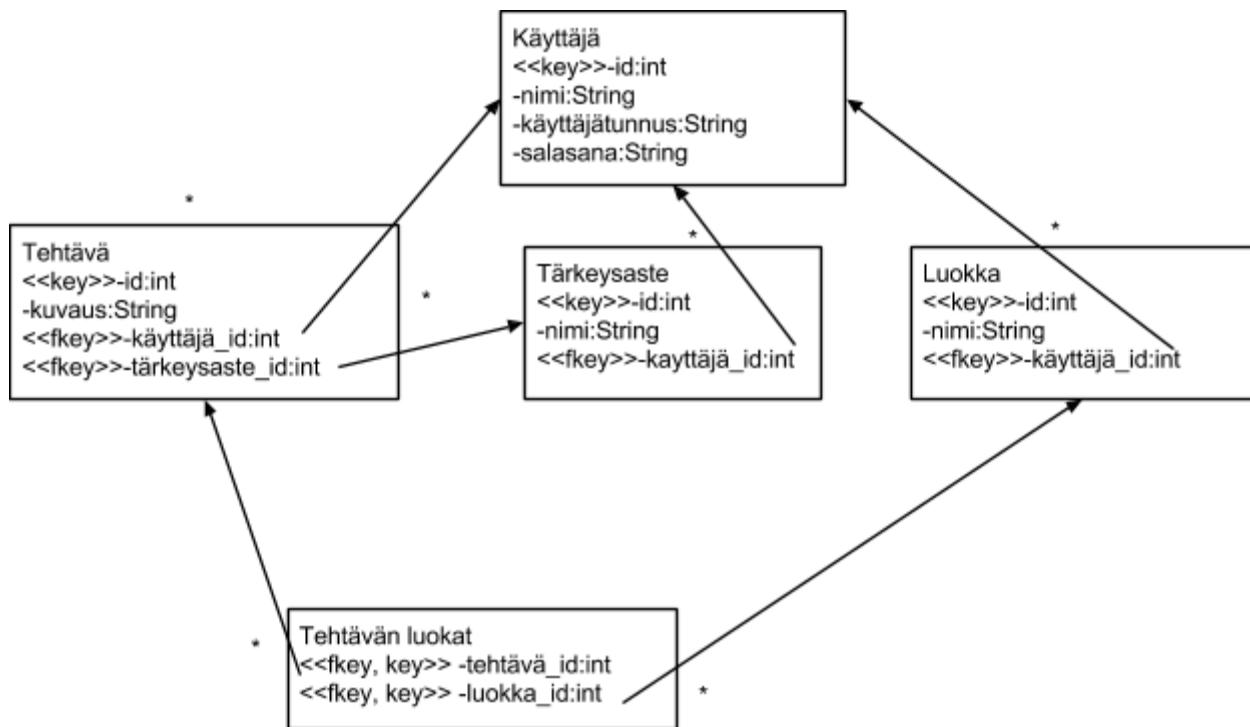
Luokka liittyy aina johonkin käyttäjään. Luokalla on nimi, joka kuvaa luokkaa jotenkin.

Tietokohde: Tehtävän luokat:

Attribuutti	Arvojoukko	Kuvailu
tehtävä_id	kokonaisluku (32-bit)	mihin tehtävään luokka kuuluu
luokka_id	kokonaisluku (32-bit)	mihin luokkaan tehtävä kuuluu

Yksi tehtävä voi kuulua useisiin luokkiin ja yhteen luokkaan voi kuulua useita tehtäviä.

5. Relaatiotietokantakaavio



SQL:n create table -lauseet annetaan tämän dokumentaation liitteenä ja lisäksi ne löytyvät osoitteesta <https://github.com/Theeta/tsoha/blob/master/sql/create-tables.sql> .

6. Järjestelmän yleisrakenne

Sovellus noudattaa MVC-mallia. Kontrollerit on sijoitettu projektin juureen, näkymät views-hakemistoon ja mallit libs/models-hakemistoon. Yleiskäyttöiset apuluokat löytyvät libs-hakemistosta. Tiedostonimet on kirjoitettu pienellä. Selkeän parin muodostavat kontrollerit ja näkymät on nimetty samalla nimellä siten, että näkymällä on _view-pääte.

7. Järjestelmän komponentit

Kontrollerit

Melkein kaikki kontrollerit käyttävät sessioon tallennettua käyttäjätietoa, vaikka sitä ei erikseen ole ehkä joka kohtaan kirjoitettu.

kayttajanmuokkaus.php

Näyttää käyttäjälle tämän omat tiedot käyttäjänmuokkausnäkymässä ja käsittlee käyttäjänmuokkausnäkymästä lähetetyt muokatut käyttäjätiedot. Tarkastaa käyttäjämallilta, että annetut tiedot ovat kelvolliset ja käskee sitten käyttäjämallin tallentaa muuttuneet tiedot tietokantaan.

kayttajanpoisto.php

Käsittlee käyttäjänmuokkausnäkymästä tulleen pyynnön poistaa kirjautunut käyttäjä. Käyttäjä kirjataan ensin ulos, jonka jälkeen kontrolleri pyytää käyttäjämallia poistamaan käyttäjän tietokannasta. Tämän jälkeen siirrytään sisäänkirjautumissivulle.

login.php

Käsittlee sisäänkirjautumisen varmistamalla, että käyttäjätunnus ja salasana vastaavat toisiaan ja löytyvät käyttäjä-tietokannasta. Kun oikea käyttäjätunnus ja salasana on annettu, käyttäjä asetetaan session käyttäjäksi ja käyttäjälle näytetään tehtävälista-näkymä. Ilman niitä käyttäjä näkee vain kirjautumisnäkymän.

logout.php

Käsittlee uloskirjautumisen poistamalla kirjautuneen käyttäjän tiedot sessiosta ja siirtymällä sisäänkirjautumissivulle.

luokanmuokkaus.php

Näyttää luokanmuokkausnäkymän kirjautuneelle käyttäjälle ja välittää tiedon kirjautuneesta käyttäjästä ja hänen luokistaan.

luokannimenmuokkaus.php

Näyttää kirjautuneelle käyttäjälle luokannimenmuokkausnäkymän ja käsittlee siellä annetut tiedot eli luokan uuden nimen. Kysyy luokkamallilta, onko nimi kelvollinen ja jos on, pyytää mallia muokkaamaan uuden nimen tietokantaan. Lopuksi siirrytään luokanmuokkausnäkymään.

luokanpoisto.php

Käsittlee luokanmuokkausnäkymän tapahtumat, joissa painetaan luokanpoistonappia. Pyytää luokkamallia poistamaan kyseisen luokan tietokannasta. Lopuksi palataan luokanmuokkausnäkymään.

tarkeysasteenmuokkaus.php

Näyttää tarkeysasteenmuokkausnäkymän kirjautuneelle käyttäjälle ja välittää myös tiedon kirjautuneesta käyttäjästä ja hänen tarkeysasteistaan.

tarkeysasteennimenmuokkaus.php

Näyttää kirjautuneelle käyttäjälle tarkeysasteennimenmuokkausnäkymän ja käsittelee tarkeysasteennimenmuokkausnäkymässä annetut tiedot eli tarkeysasteen uuden nimen. Kysyy tarkeysaste-mallilta, onko nimi kelvollinen ja jos on, pyytää mallia päivittämään sen tietokantaan. Lopuksi siirrytään tarkeysasteenmuokkausnäkymään.

tarkeysasteenpoisto.php

Käsittelee tarkeysasteenmuokkausnäkymän tapahtumat, joissa painetaan tarkeysasteen poisto -nappia. Pyytää tarkeysastemallia poistamaan kyseisen tarkeysasteen tietokannasta. Lopuksi siirrytään takaisin tarkeysasteenmuokkausnäkymään.

tehtavalista.php

Käsittelee tehtävälistanäkymän tapahtumat, joissa vaihdetaan sivua tai haetaan tehtäviä tietyyn tarkeysasteen ja/tai luokan perusteella. Täällä voi myös vaihtaa sitä, kuinka monta tehtävää yhdellä sivulla näytetään. Hakee tehtävämallilta annettujen hakuehtojen (sivunumero, tarkeysaste, luokka) mukaisen listan tehtäviä ja antaa sen tehtävälistanäkymälle, jos käyttäjä on kirjautunut sisään.

tehtavanmuokkaus.php

Huolehtii tehtävänmuokkausnäkymän esittämisestä. Saa tiedon siitä, mitä tehtävää käyttäjä haluaa muokata ja hakee tämän tehtävän tehtävämallilta ja välittää sen tehtävämuokkausnäkymälle. Käsittelee tehtävämuokkausnäkymässä annetut kyseistä tehtävää muokkaavat tiedot. Tarkastaa mallilta, että tarvittavat parametrit on annettu ja käskee mallin sitten muokata tietokantaa. Onnistuneen muokkauksen jälkeen siirrytään takaisin tehtävälistaan.

tehtavanpoisto.php

Huolehtii tehtävien poistosta, jos käyttäjä painaa tehtävänpoistonappia tehtävälistanäkymässä. Pyytää tehtävämallia poistamaan kyseisen tehtävän sen id:n perusteella tietokannasta ja siirtää käyttäjän lopuksi takaisin tehtävälistaan.

uusikayttaja.php

Käsittelee uusikäyttäjänäkymästä tulevat uuden käyttäjän luontipyynnöt. Tarkastaa käyttäjämallilta niiden kelvoisuuden ja jos ne ovat kunnossa, pyytää käyttäjämallia lisäämään uuden käyttäjän tietokantaan. Tämän jälkeen siirrytään sisäänkirjautumissivulle.

uusiluokka.php

Käsittelee luokanmuokkausnäkymästä tulevat uuden luokan luontipyynnöt. Tarkastaa luokkamallilta, kelpaako uusi luokka, ja jos kelpaa, pyytää mallia lisäämään sen tietokantaan. Lopuksi palataan luokanmuokkausnäkymään.

uusitarkeysaste.php

Käsittelee tärkeysasteenmuokkausnäkymästä tulevat uuden tärkeysasteen luotipyynnöt. Tarkastaa tärkeysastemallilta, kelpaako uusi tärkeysaste ja jos kelpaa, pyytää tärkeysastemallia lisäämään tietokantaan sen nimisen tärkeysasteen.

uusitehtava.php

Kysyy tehtävämallilta kyseisen käyttäjän tärkeysasteet ja välittää ne uusitehtävänäkymälle. Ottaa vastaan uusitehtävänäkymästä tulevat tiedot uudesta tehtävästä ja tarkastaa tehtävämallilta niiden kelvollisuuden. Jos ne ovat kunnossa, pyytää tehtävämallia lisäämään tietokantaan kyseisen uuden tehtävän. Onnistuneen tehtävänlisäyksen jälkeen siirtyy tehtävälistänäkymään.

Näkymät views/

kayttajanmuokkaus_view.php

Näyttää lomakkeen, jonka avulla voi käyttäjä voi muokata nimeä tai salasanaa.

kirjauduensinulos.php

Virheilmoitussivu, jolta löytyy myös linkki uloskirjautumista varten.

kirjautuminen.php

Näyttää sisäänkirjautumislomakkeen, johon voi syöttää käyttäjätunnuksen ja salasanan.

luokanmuokkaus_view.php

Esittää käyttäjän luokat taulukkomuodossa. Jokaisen luokan kohdalla on nappi, josta luokan voi poistaa ja toinen nappi, josta siirtyy sivulle, jossa on mahdollista muokata luokan nimeä. Lisäksi sivulla on lomake, jonka avulla voi lisätä uuden luokan.

luokannimenmuokkaus_view.php

Näyttää lomakkeen, jolla on mahdollista muuttaa kyseisen luokan nimeä.

pohja.php

Näyttää kaikilla sivuilla olevan yläpalkin. Sisäänkirjautumisen jälkeen palkista löytyy linkit tehtävälistaan, uuden tehtävän lisäykseen, tärkeysasteiden muokkaukseen ja uloskirjautumiseen. Huolehtii myös mahdollisten sessioon tallennettujen ilmoitusten ja dataaulukkoon tallennettujen virheiden näyttämisestä.

tarkeysasteenmuokkaus_view.php

Esittää käyttäjän tärkeysasteet taulukkomuodossa. Jokaisen tärkeysasteen kohdalla on nappi, josta tärkeysasteen voi poistaa ja toinen nappi, josta siirtyy sivulle, jossa on mahdollista muokata tärkeysasteen nimeä. Lisäksi sivulla on lomake, jonka avulla voi lisätä uuden tärkeysasteen.

`tarkeysasteennimenmuokkaus_view.php`

Näyttää lomakkeen, jolla on mahdollista muuttaa kyseisen tärkeysasteen nimeä.

`tehtavalista_view.php`

Näyttää käyttäjän tehtävät taulukkomuodossa. Jokaisen tehtävän kohdalla on myös nappi, josta tehtävän voi poistaa ja toinen josta voi siirtyä kyseisen tehtävän muokkaussivulle. Lisäksi näkymässä on nappulat, joiden avulla voi vaihtaa sivua, jos tehtäviä on useampia sivuja. Näkymässä on myös valikko, josta voi valita vain tietyt tärkeysasteen tehtävät.

`tehtavanmuokkaus_view.php`

Näyttää lomakkeen, jonka avulla voi muokata olemassa olevaa tehtävää eli vaihtaa sen tärkeysastetta tai kuvausta.

`uusikayttaja_view.php`

Näyttää lomakkeen, jonka avulla voi luoda uuden käyttäjän.

`uusitehtava_view.php`

Näyttää lomakkeen, jonka avulla voi luoda uuden tehtävän eli antaa tehtävälle kuvaksen ja tärkeysasteen.

`vaarahenkilo.php`

Virheilmoitussivu, joka sisältää linkin takaisin käyttäjän tehtävälistaan. Näytetään, jos käyttäjä yrittää tehdä jotain kiellettyä.

Tietokantataulut sql/

`add-test-data.sql`

Sisältää testaamiseen käytettyä tietokantoihin syötettävää dataa. Tällä hetkellä esimerkiksi uusia käyttäjiä saa vain lisäämällä tiedot tänne.

`create-tables.sql`

Sisältää tietokantataulujen luomiseen tarvittavat `create table` -lauseet.

`drop-tables.sql`

Sisältää tietokantataulujen poistoon tarvittavat lauseet.

Yleiskäytöiset tiedostot libs/

common.php

Sisältää yleiskäytöisiä funktioita, joiden avulla esimerkiksi tarkastetaan, onko käyttäjä kirjautunut, tai siirrytään tietylle sivulle tai näytetään tietty näkymä.

config.php

Sisältää täältä: <http://stackoverflow.com/questions/1475297/phps-white-screen-of-death>
kopioidun virheidenkäsittelyn, jota on käytetty apuna koodissa olevien virheiden etsimisessä.
Ilman tästä virheet johtavat lähes aina valkoiseen sivuun.

tietokantayhteys.php

Sisältää tietokantayteyden luomista varten tarvittavat tiedot ja luo yhteyden.

Mallit libs/models/

kayttaja.php

Sisältää käyttäjän getterit, setterit, konstruktorin ja funktiot, joiden avulla tietokannasta voi hakea tietyn käyttäjän ja tarkastaa, onko käyttäjätunnus jo varattu. Sisältää myös funktiot, joilla käyttäjätauluun voi lisätä uuden käyttäjän, poistaa käyttäjän tai muokata olemassa olevaa käyttäjää. Lisäksi sisältää funktion, jolla voi tarkastaa uuden käyttäjän kelvollisuuden.

luokka.php

Sisältää luokan getterit, setterit, konstruktorin ja funktiot, joiden avulla tietokannasta voi etsiä luokan sen id:n perusteella tai käyttäjän id:n perusteella. Sisältää myös funktiot, joilla luokkatauluun voi lisätä uuden luokan, poistaa luokan tai muokata olemassa olevaa luokkaa. Lisäksi sisältää funktion, jolla voi tarkastaa uuden luokan kelvollisuuden.

tarkeysaste.php

Sisältää tärkeysasteen getterit, setterit, konstruktorin ja funktiot, joiden avulla tietokannasta voi etsiä tärkeysasteen sen id:n perusteella tai käyttäjän id:n perusteella. Sisältää myös funktiot, joilla tärkeysastetauluun voi lisätä uuden tärkeysasteen, poistaa tärkeysasteen tai muokata olemassa olevaa tärkeysastetta. Lisäksi sisältää funktion, jolla voi tarkastaa uuden tärkeysasteen kelvollisuuden.

tehtava.php

Sisältää tehtävän getterit, setterit, konstruktorin ja funktiot, joiden avulla tietokannasta voi etsiä tehtävän sen id:n perusteella tai tehtävät käyttäjän id:n perusteella. Sisältää myös funktiot, joilla tehtävätauluun voi lisätä uuden tehtävän, poistaa tehtävän tai muokata olemassa olevaa tehtävää. Lisäksi sisältää funktion, jolla voi tarkastaa uuden tehtävän kelvollisuuden ja funktiot

joilla voi laskea käyttäjän tehtävien lukumäärän tai käyttäjän tiettyyn tärkeysasteesseen liittyvien tehtävien lukumäärän.

`tehtavanluokat.php`

Sisältää tehtäväluokat-taulun getterit ja setterit.

JavaScript js/

`bootstrap.js`

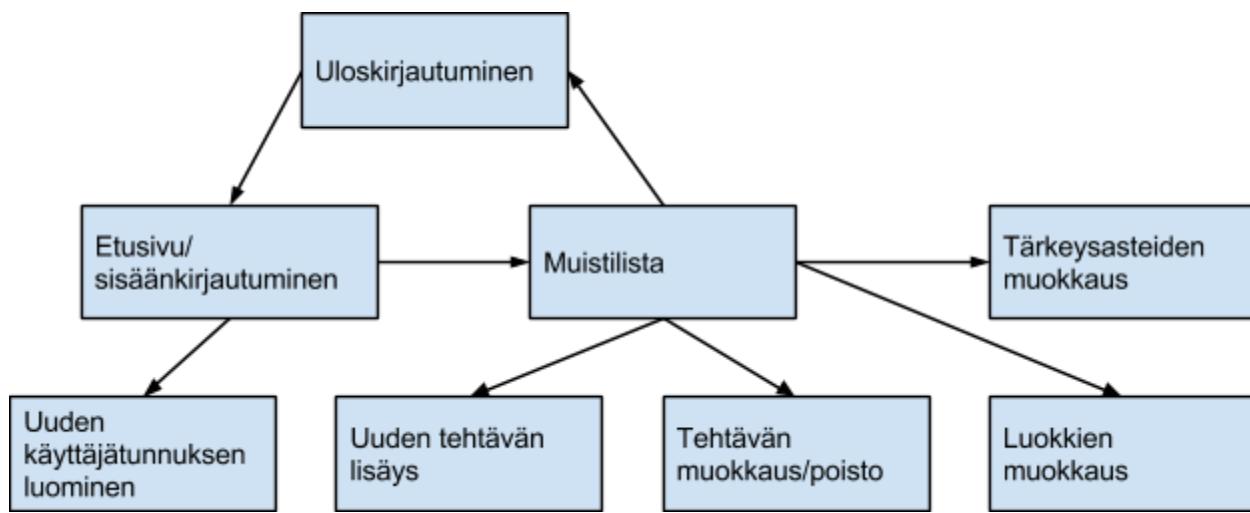
Tarvitaan, jotta tehtävälistänäkymässä oleva pudotusvalikko toimii.

css-tyylitiedostot css/

`bootstrap.css`

Parantaa joidenkin näkymien ulkonäköä.

8. Käyttöliittymä



Uuden käyttäjätunnuksen luonti -sivulta palataan sisäänskirjautumissivulle. Muilta sivuilta, joista ei lähde nuolia, palataan muistilista-sivulle eli nuolet ovat kaksisuuntaisia kaikissa muissa tapauksissa, paitsi uloskirjautumisessa. Uloskirjautumisen jälkeen palataan sisäänskirjautumissivulle. Ilman sisäänskirjautumista näkee vain sisäänskirjautumissivun ja uuden käyttäjätunnuksen luonti -sivun. Sisäänskirjautunut käyttäjä näkee jatkuvasti navigointipalkin, jonka avulla pääsee kaikille sisäänskirjautuneelle käyttäjälle tarkoitetuille sivuille eli kaikkialle paitsi uuden käyttäjätunnuksen luontiin ja sisäänskirjautumiseen.

9. Asennustiedot

Sovellus on nyt tallennettu laitoksen users-palvelimen htdocs-hakemistoon, missä se on näkyvissä nettiin. Sovelluksen voi asentaa kopioimalla tiedostot kyseisestä paikasta esimerkiksi omaan htdocs-hakemistoonsa. Sovellus vaatii PostgreSQL-tietokannan, jonka saa laitoksen usersilla käyttöön skriptin (wanna-postgres) avulla. Lopuksi tietokannan yhteystiedot on muokattava oikeaksi tiedostoon libs/tietokantayhteys.php.

10. Käyttöohje

Sovelluksen sisäänkirjautuminen löytyy osoitteesta

<http://imberg.users.cs.helsinki.fi/Muistilista/login.php>. Sovellusta voi testata kirjautumalla sisään seuraavilla tiedoilla:

Käyttäjätunnus: testi

Salasana: salasana

Sovellukseen voi myös luoda oman käyttäjän oikeassa yläkulmassa olevasta Uusi käyttäjä -linkistä. Lomakkeen kaikki kentät on täytettävä kelvollisilla tiedoilla, minkä jälkeen kyseinen käyttäjä on luotu ja voi kirjautua sisään.

Sisäänkirjautumisen jälkeen voi tarkastella käyttäjän tehtävälistaa. Tehtäviä voi lisätä yläreunassa olevasta Lisää uusi tehtävä -napista. Tärkeysasteita voi tarkastella, lisätä, muokata ja poistaa Muokkaa tärkeysasteita -linkistä. Vastaavasti luokkia muokataan Muokkaa luokkia -linkistä. Käyttäjätietoja voi muokata tai käyttäjän voi poistaa Käyttäjätiedot-linkistä. Sen vieressä olevasta Kirjaudu ulos -linkistä voi kirjautua ulos. Valitse tärkeysaste - ja Valitse luokka -pudotusvalikoista voi valita näytettäväksi vain tiettyyn tärkeyssteeeseen ja/tai luokkaan kuuluvat tehtävät.

11. Testaus, puutteet ja jatkokehitysideoat

Ohjelmaa on testattu vain kokeilemalla sen toimintaa erilaisilla syötteillä ja muuttamalla annettavia/ lähetettäviä parametrejä. Ohjelmalla ei ole testejä, eikä sitä ole testattu systemaattisesti. Koodikatselmoinnin yhteydessä myös ulkopuoliset testasivat ohjelmaa ja sitä on kehitetty saadun palautteen perusteella.

Ohjelman puutteet johtuvat lähinnä suunnitteluvaiheen puutteista. Ohjelman tärkeysasteiden kannattaisi olla lukuarvoja, jotta tehtävät voisi lajitella tärkeysasteen mukaiseen järjestykseen. Lisäksi tehtävissä voisi esimerkiksi olla mahdollisuus merkitä ne tehdyiksi tai tekemättöviksi, jolloin tehtyjä tehtäviä ei tarvitsisi poistaa tietokannasta. Tehtävällä voi tällä hetkellä olla useampia luokkia, mutta kaikki luokat ovat samanarvoisia. Luokille voisi lisätä yläluokka-ominaisuuden, jolloin luokista saisi lisäksi sisäkkäisiä. Lisäksi tehtävillä voisi olla kuvauksen lisäksi nimi ja luokilla taas nimen lisäksi myös kuvaus. Käyttäjän nimi on tällä hetkellä jokseenkin turha, joten sen voisi jättää pois tai pyrkiä käyttämään sitä jossakin.

Ohjelmassa on myös jonkin verran toistoa, josta olisi kyllä päässyt helposti eroon, mutta koska pienikin muutos vaatii niin paljon testaamista, en uskaltanut lähteä kovin paljon koodia muokkaamaan, sen jälkeen kun sain kaiken toimimaan. Jos kurssi olisi kestänyt vielä toisenkin periodin, olisi voinut toteuttaa myös tällaisia muutoksia.

12. Omat kokemukset

Sovelluksen tekoon oli varattu niin vähän aikaa, että jatkuvasti piti todella miettiä, mitä kannattaa edes yrittää toteuttaa. Tiesin kyllä, ettei harjoitustyökurssien kanssa kannattaisi ottaa samanaikaisesti muita kursseja, mutta koska tästä kurssista ei saa 15 opintopistettä, oli muitakin kursseja pakko ottaa sopivasti, jotta pääsi lähemmäs tuota vaadittua opintopistemäärää. Tällä kurssilla tuntikirjanpito olisi ollut mielenkiintoista, sillä tuntuu että kurssiin sai käyttää lopputtomasti aikaa, mutta silti lopputulos oli hyvin yksinkertainen. Aikaa kului hyvin paljon myös dokumentin kirjoittamiseen, kuvien piirtelyyn, koodikatselmiin ja muuhun sellaiseen. Näissäkin kaikissa olisi pitänyt varmaan vain katsoa, mistä voi saada eniten pisteytä, tehdä se ja unohtaa muu. Voisiko esimerkiksi tästä dokumentoinnista karsia muutaman kohdan, jotta varsinaiselle koodaamiselle jäisi edes pari tuntia enemmän aikaa.

Opin kyllä todella paljon tästä työstä ja oli hauskaa, kun jossain vaiheessa vain ymmärsi kuinka kaikki olisikin järkevää toteuttaa. Harmi ettei aika riittänyt koko työn toteuttamiseen uudestaan alusta alkaen.

13. Liitteet

Tehtävänkuvaus löytyy täältä: <http://advancedkittenry.github.io/suunnittelu/aiheet/Muistilista.html>

Ohjelmakoodia en viitsinyt kopioida tähän, koska en jaksa uskoa, että kukaan haluaa sitä tästä yhtenä pötkönä lukea. Se löytyy kuitenkin täältä: <https://github.com/Theeta/tsoha>

Create table lauseet löytyvät täältä:

<https://github.com/Theeta/tsoha/blob/master/sql/create-tables.sql>

ja koska ovat sen verran lyhyet, niin lisäksi tästä:

```
CREATE TABLE kayttaja
(
    id serial primary key,
    nimi varchar(50),
    tunnus varchar(20),
    salasana varchar(20)
);

CREATE TABLE luokka
(
    id serial primary key,
    nimi varchar(255),
    kayttaja_id integer references kayttaja(id) on delete cascade on update cascade
);

CREATE TABLE tarkeysaste
(
    id serial primary key,
    nimi varchar(255),
    kayttaja_id integer references kayttaja(id) on delete cascade on update cascade
);

CREATE TABLE tehtava
(
    id serial primary key,
    kuvaus varchar(1000),
    kayttaja_id integer references kayttaja(id) on delete cascade on update cascade,
```

```
tarkeysaste_id integer references tarkeysaste(id) on delete cascade on update cascade
);

CREATE TABLE tehtavanluokat
(
tehtava_id integer references tehtava(id) on delete cascade on update cascade,
luokka_id integer references luokka(id) on delete cascade on update cascade,
primary key(tehtava_id, luokka_id)
);
```