

Projektinhallintajärjestelmä

Toteutusdokumentti

Markus Korpinen

11. maaliskuuta 2013

Johdanto

Toteutetaan projektinhallintajärjestelmä, jonka avulla voidaan seurata eri projektien etene- mistä. Käyttäjät voivat kirjautumisen jälkeen luoda omia projekteja ja jakaa niitä muiden käyttäjien kanssa. Projektien alle voidaan asettaa tehtäviä eri prioriteeteilla ja aikatauluilla. Projektinhallintajärjestelmä toteutetaan käyttäen Ruby-kieltä ja MVC-malliin pohjautuvaa Ruby on Rails -ohjelmistokehystä. Tietokantannaksi valitaan PostgreSQL ja Web-palvelimeksi WebRICK. Versionhallintana käytetään gitia ja ohjelmisto julkaistaan Heroku.com palvelussa.

Rajaukset

Lisättiin tietokantaan *sessions*-taulu, joka huolehtii käyttäjien järjestelmään kirjautumisesta.

Ohjelmiston yleisrakenne

Kuten kaikki Rails-ohjelmistot, järjestelmä käyttää MVC-arkkitehtuuria, eli koostuu malleista (Models), näkymistä (Views) ja käsittelijöistä (Controllers).

Näkymät kuvaavat järjestelmän tiedon tallentamisen ja muun käsittelyn, eli vastaavat tietokannan tauluja. Näkymät vastaavat tietojen esittämisestä käyttöliittymässä. Käsittelijät taas vastaanottavat käyttäjältä tulevat pyynnöt ja hakevat tarvittavat tiedot mallilta ja lähettävät ne näkymille.

Järjestelmän komponentit

Mallit (models)

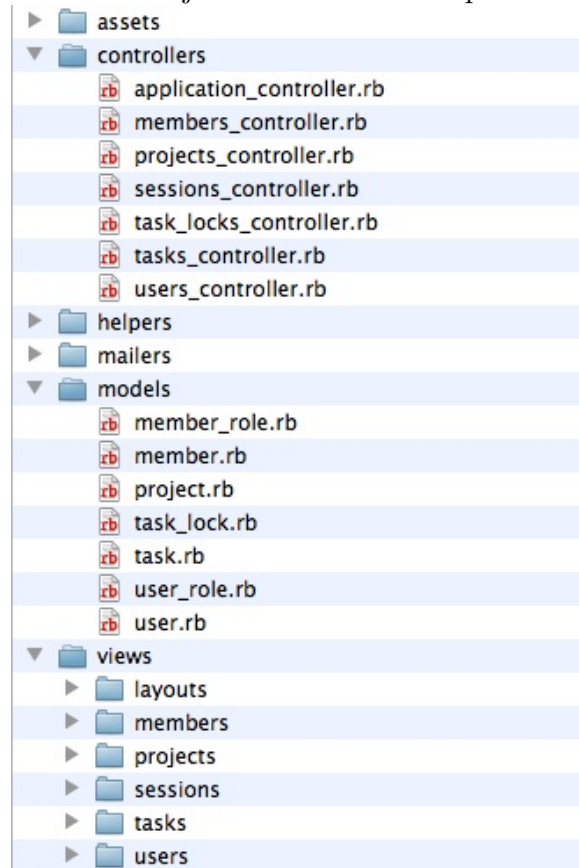
Mallin jälkeen suluissa mallia vastaavan tietokantataulun kentät. Kaikissa tauluissa mainittujen kenttien lisäksi kentät `created_at:datetime` ja `updated_at:datetime`.

Member

(`project_id:int`, `user_id:int`, `member_role_id:int`, `title:string`)

Tauluun tallennetaan kunkin projektin jäsenet. Viittaukset projektiin, käyttäjään ja jäsenrooliin.

Kuva 1: Järjestelmän MVC-komponentit



MemberRole

(role:string)

Taulussa tallennettuna projektin jäsenen roolit, kuten "creator" tai "user".

Project

(name:string, url:string, description:text, visibility:string, deadline:datetime)

Vaaditut kentät ovat name ja deadline. Mallissa huolehditaan, että projektia poistettaessa poistetaan samalla kaikki projektin tehtävät ja jäsenet.

Task

(project_id:int, description:text, priority:int, status:string, deadline:datetime, comment:string, name:string)

Tehtävää poistettaessa poistetaan samalla kaikki tehtävään liittyvät lukot (task_locks). Tehtävässä viittaus projektiin.

TaskLock

(task_id:int, user_id:int)

Viittaukset tehtävään ja käyttäjään. Projektin hallinnoijan allokoidessa tehtävän projektin jäsenelle, luodaan uusi rivi task.locks-tilaan.

User

(user_role_id:int, username:string, email:string, password_salt:string, encrypted_password:string)

Malli huolehtii uuden käyttäjän luonnista tietokantaan. Salasana kryptataan suolattuna SHA2-algoritmia käyttäen.

UserRole

(role:string)

Taulussa tallennettuna järjestelmän käyttäjien roolit, kuten "administrator".

Käsittelijät (controllers)

Käsittelijän jälkeen suluissa käsittelijän metodit (protected tai private hakasuluissa).

ApplicationController

([logged_in?], [authorize], [is_admin?], [can_add?])

Käsittelijän metodit tarkistavat käyttäjän sisäänkirjautumisen tilan ja huolehtivat käyttöoikeuksien myöntämisestä riippuen siitä, onko käyttäjä jonkin projektin jäsen, tai onko käyttäjällä hallintaoikeudet projektiin.

MembersController

(index, add, new, create)

Index-metodia lukuunottamatta ApplicationController#can_add?-metodi tarkastaa ennen käsittelijän metodikutsuja voiko käyttäjä lisätä uusia jäseniä projektiin, eli onko käyttäjä joko projektin luoja tai hallinnoija. Add-metodi luo instanssimuuttujan @users, jossa on kaikki järjestelmän käyttäjät, jotka eivät vielä ole projektin jäseniä. Tämä muuttuja välitetään vastaavalle näkymälle, josta käyttäjä voi valita jäseneksi lisättävän käyttäjän.

ProjectsController

(index, show, new, edit, create, update, destroy, [is_creator?], [can_access?], [can_modify?])

Ennen käsittelijän metodikutsuja tarkistetaan onko käyttäjä kirjautunut sisään (ApplicationController#authorize, lukuunottamatta #index- ja #show-metodeja), onko käyttäjällä pääsy projektiin (#can_access?, paitsi #index, #new ja #create), onko käyttäjällä oikeus muokata projektia (#can_modify?, ainoastaan #edit, #update ja #destroy) ja onko käyttäjä projektin luoja (#is_creator?, vain tuhohtaessa projekti, eli #destroy). #index-metodi hakee kaikki järjestelmän julkiset projektit ja sitä vastaava näkymä listaa ne. #show-metodissa haetaan näytettävän projektin lisäksi kaikki projektin tehtävät ja jäsenet.

SessionsController

(new, create, destroy)

Käsittelijä vastaa käyttäjän sisäänkirjautumisesta. `#create`-metodi luo uuden session, mikäli käyttäjän tunnistautumistiedot ovat oikein, muutoin ohjaa takaisin kirjautumissivulle ilmoittaen epävalidista käyttäjätunnuksesta tai salasanasta. `#destroy`-metodi resetoit session tiedot, jolloin käyttäjä on kirjautunut ulos järjestelmästä.

TaskLocksController

(new, create)

Ennen käsittelijän metodikutsuja tarkistetaan `ApplicationController#can_add?`-metodilta, onko käyttäjällä oikeus lisätä projektin tehtävälle uusia lukkoja.

TasksController

(show, new, edit, create, update, destroy) Kaikille käsittelijän metodikutsuille (paitsi `#show`) tarkistetaan onko käyttäjä kirjautunut järjestelmään (`ApplicationController#authorize`), ja onko käyttäjällä oikeus luoda uusia tehtäviä projektiin ja muokata niitä (`ApplicationController#can_add?`).

UsersController

(index, show, new, edit, create, update, destroy, [`own_account?`])

Järjestelmään sisäänkirjautumaton käyttäjä voi luoda uuden käyttäjätilin (`#new`- ja `#create`-metodit), muille käsittelijän metodikutsuille tehdään tarkistukset: `ApplicationController#is_admin?` tarkistaa ennen `#index`-metodia onko käyttäjä järjestelmän hallinnoija. `#own_account?` tarkistaa ennen `#show`-, `#edit`-, `#update`- ja `#destroy`-kutsuja onko käyttäjä tarkastelemassa omaa käyttäjätiliään, jolloin näihin metodeihin sallitaan pääsy.

Näkymät (views)

Kaikki näkymätiedostot vastaavat kunkin käsittelijän samannimistä metodia, ja metodissa määritellyt instanssimuuttujat tai palautusarvot ovat näkymän käytettävissä. Samoin kaikki esim. lomakkeella lähetetyt tiedot annetaan metodille käsiteltäväksi erityisessä `'params'`-hajautustaulussa.

layouts/

(`application.html.erb`)

Sisältää käyttöliittymän navigaation ja linkit sisäänkirjautumiseen tai käyttäjätilin luontiin, riippuen siitä, onko käyttäjä kirjautunut järjestelmään vai ei. Navigaatiossa on linkit projekteihin (sivun generoiva metodi: `ProjectsController#index`), ja mikäli sisäänkirjautunut käyttäjä on järjestelmän hallinnoija, myös käyttäjiin (`UsersController#index`).

members/

(_form.html.erb, add.html.erb, index.html.erb, new.html.erb)

Form on lomake, jota käytetään uuden jäsenen lisäämiseksi projektiin new.html.erb:n välityksellä. Lomakkeen kentät ovat 'title', joka on jäsenen titteli projektissa ja 'member_role_id', johon valitaan pudotusvalikosta jäsenrooli. Piilotettuna kenttänä on 'user_id', jonka arvo tulee #add-metodille annetusta käyttäjän id:stä. add.html.erb sisältää listan kaikista mahdollisista järjestelmän käyttäjistä, jotka voidaan lisätä projektiin. index.html.erb sisältää listauksen kaikista projektin jäsenistä, ja jos sisäänkirjautunut käyttäjä on projektin hallinnoija, hänelle näytetään linkki uuden jäsenen lisäämiseen.

projects/

(_form.html.erb, edit.html.erb, index.html.erb, new.html.erb, show.html.erb)

Form on lomake, jota käytetään uuden projektin lisäämiseen ja muokkaamiseen new.html.erb:n ja edit.html.erb:n välityksellä. Lomakkeen kentät ovat 'name', projektin nimi, 'description', projektikuvaus, 'url', linkki projektin nettisivuun, 'deadline' jossa päivämäärää varten pudotusvalikko ja 'visibility' jossa valitaan painikkeella joko 'Public' tai 'Private'. Mikäli projektia luotaessa/päivitetessä tapahtuu virheitä (vaadittujen tietojen kentät ovat tyhjiä), näytetään käyttäjälle virheilmoitukset.

index.html.erb näyttää kaikki ProjectsController#index-metodissa muuttujalle @projects annetut projektit, eli kaikki julkiseksi tallennetut projektit. Kunkin projektin kohdalla on linkki projektiin (ProjectsController#show), ja jos käyttäjä on projektin hallinnoija, myös linkit projektin muokkaamiseen (ProjectsController#edit) ja tuhoamiseen (ProjectsController#destroy). Lisäksi on linkki uuden projektin luomiseen (ProjectsController#new).

show.html.erb näyttää kaikki projektin tiedot, sekä projektin jäsenet ja tehtävät. Sivulla on linkki kaikkien projektin jäsenten tarkastelemiseen (MembersController#index), ja kunkin tehtävän kohdalla linkki tehtävään (TasksController#show). Projektin hallinnoijille näytetään myös linkit uuden tehtävän lisäämiseen (TasksController#new) ja projektin muokkaamiseen ja tuhoamiseen (ProjectsController, #edit ja #destroy).

sessions/

(create.html.erb, destroy.html.erb, new.html.erb)

new.html.erb tarjoaa käyttäjälle lomakkeen sisäänkirjautumista varten. Lomakkeen kentät ovat 'username' ja 'password', jossa kirjoitettu teksti ei näy selväkielisenä. Lomakkeen tiedot lähetetään poikkeuksellisesti SessionsController#create-metodille.

tasks/

(_form.html.erb, edit.html.erb, new.html.erb, show.html.erb)

Form on lomake, jota käytetään uusien tehtävien lisäämiseen (näkyvässä new.html.erb), tai tehtävien muokkaamiseen (edit.html.erb). Lomakkeen kentät ovat 'name', 'description', 'priority', jossa valitaan pudotusvalikosta tehtävän prioriteetti väliltä 1..5, 'status', jossa valitaan tehtävän tila pudotusvalikosta, jossa vaihtoehtoina ovat 'pending' ja 'completed', 'deadline', jossa arvo valitaan päivämäärävalikosta ja 'comment', joka on tekstikenttä. show.html.erb näyttää kaikki tehtävän tiedot, ja mikäli tehtävä on allokoitu joillekin jäsenille, myös näiden jäsenten käyttäjänimet.

Projektin hallinnoijille näytetään lisäksi lomake, jolla voi allokoida tehtävän uudelle jäsenelle. Lomakkeessa valitaan pudotusvalikosta käyttäjänimi projektin jäsenten joukosta, ja lähetetään tiedot `TaskLocksController#create`-metodille. Projektin hallinnoijille näytetään lisäksi linkit tehtävän muokkaamiseen ja tuhoamiseen (`TasksController`, `#edit` ja `#destroy`).

users/

(`_form.html.erb`, `edit.html.erb`, `index.html.erb`, `new.html.erb`, `show.html.erb`)

Form on lomake, jota käytetään uusien jäsenten lisäämiseen (näkyvässä `new.html.erb`), tai jäsenten muokkaamiseen (`edit.html.erb`). Lomakkeen kentät ovat `'username'`, `'email'`, `'password'` ja `'password_confirmation'`, joista kaksi viimeiseksi mainittua eivät näytä niihin kirjoitettua tekstiä selväkielisenä. Sivu näyttää myös virheet syötteissä, mikäli sellaisia on (esim. `'password'` ja `'password_confirmation'` eroavat toisistaan). `index.html.erb` näyttää kaikki järjestelmän käyttäjät taulukkona. Sivulla on myös linkit käyttäjän sivun katseluun, käyttäjän muokkaamiseen ja tuhoamiseen, sekä uuden käyttäjän lisäämiseen (`UserController`, `#show`, `#edit`, `#destroy` ja `#new`). `show.html.erb` näyttää käyttäjän tiedot, sekä listaa linkit projekteihin joissa käyttäjä on jäsenenä (`ProjectsController#show`). Lisäksi sivulla on linkki käyttäjän tietojen muokkaamiseen (`UserController#edit`).

Asennustiedot

Ohjelmisto noudattaa tyypillisiä Rails-käytäntöjä hakemistorakenteensa osalta. Käytetty Ruby-versio on 1.9.3 ja Rails-versio 3.2.11.

Käynnistys- / käyttöohje

Ohjelmisto on saavutettavissa osoitteessa: <http://shrouded-falls-4384.herokuapp.com>

Liitteet

`tsohar/doc/tsohar.sql` (tietokannan määrittelevät `CREATE TABLE` -lauseet)

`tsohar/doc/suunnitteludokumentti.pdf`

lähdekoodi: <https://github.com/zillakot/tsohar>

sovellus: <http://shrouded-falls-4384.herokuapp.com>