Sovelluksen Ohtu-lukuvinkit arkkitehtuuri

Team Team
https://github.com/luupanu/ohtu-lukuvinkit

Sisällysluettelo

Yleiskuvaus arkkitehtuurista

Sovelluksemme *ohtu-lukuvinkit* tai *Reading tips* on lukuvinkkien organisoitiin tarkoitettu ohjelma. Ohjelma on toteutettu web-sovelluksena Javalla käyttäen Spring-frameworkia sekä Thymeleafia näkymien luomisen apuna. Valitsimme nämä teknologiat kompromissina, sillä Java oli käytännössä ainoa kieli jota kaikki tiimimme jäsenet osasivat.

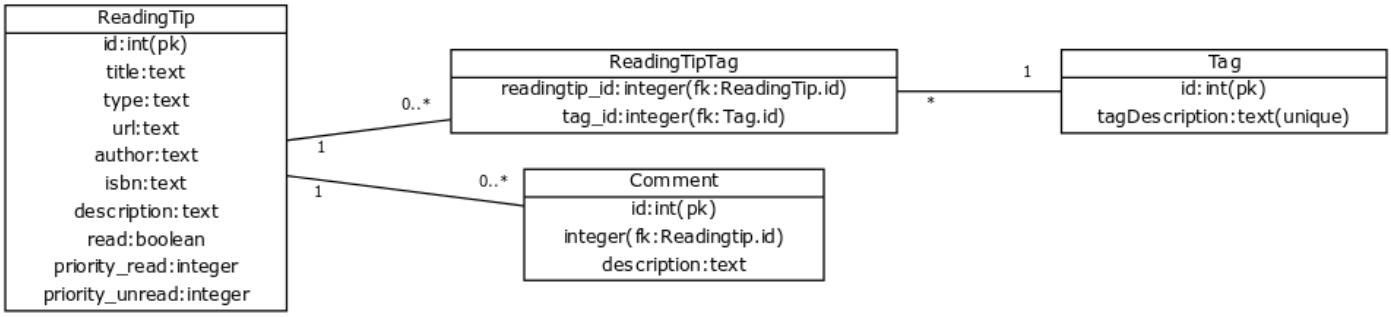
Projektimme on toteutettu niin kutsuttuna *single-page applicationina*, eli siis käyttäjä sovelluksen avattuaan löytää kaiken toiminnallisuuden heti ensimmäiseltä sivulta. Saadaksemme tämän onnistumaan saumattomasti sovelluksemme käyttää hyödykseen Javascriptiä, ja käyttäjän selaimen tuleekin tukea tätä.

Lukuvinkit tallentuvat paikalliseen relaatiotietokantaan. Valitsimme tietokantajärjestelmäksemme kevyen sqlite3:n.

Projektimme pyrkii mahdollisimman hyvin noudattamaan MVC-mallia. Javan perinteiden mukaan käytämme tietokannan abstrahoimiseen DAO-mallia.

Tietokanta

Tietokantakaavio



Tietokannassa tällä hetkellä ReadingTip kuvaa lukuvinkkiä. Lukuvinkkejä on kolmea eri tyyppiä (Article, Book, Link), jota kuvastaa attribuutti *type*. Artikkeleilla on attribuutti *author*, kirjoilla *author* ja *isbn* sekä linkeillä *url*. Kaikilla kolmella on myös attribuutit *title* ja *description*. Pidämme tällä hetkellä näitä kaikkia kolmea samassa tietokantataulussa, mikä rikkoo periaatetta single responsibility principle. Teimme tämän tietoisena päätöksenä siksi, että tällä hetkellä tietokannan muuttaminen olisi liian kallis operaatio toteutettavaksi. Attribuutti *read* kuvastaa, onko lukuvinkki jo luettu, jolloin se näkyy sovelluksessa erilaisena. Attribuutit *priority_read* ja *priority_unread* kuvastavat lukuvinkkien prioriteettia. Koska luetut vinkit halutaan erottaa lukemattomista, päädyimme toteuttamaan priorisoinnin jakamalla sen kahteen eri attribuuttiin: lukuvinkeille, joita ei ole luettu ja lukuvinkeille, jotka on merkattu luetuiksi.

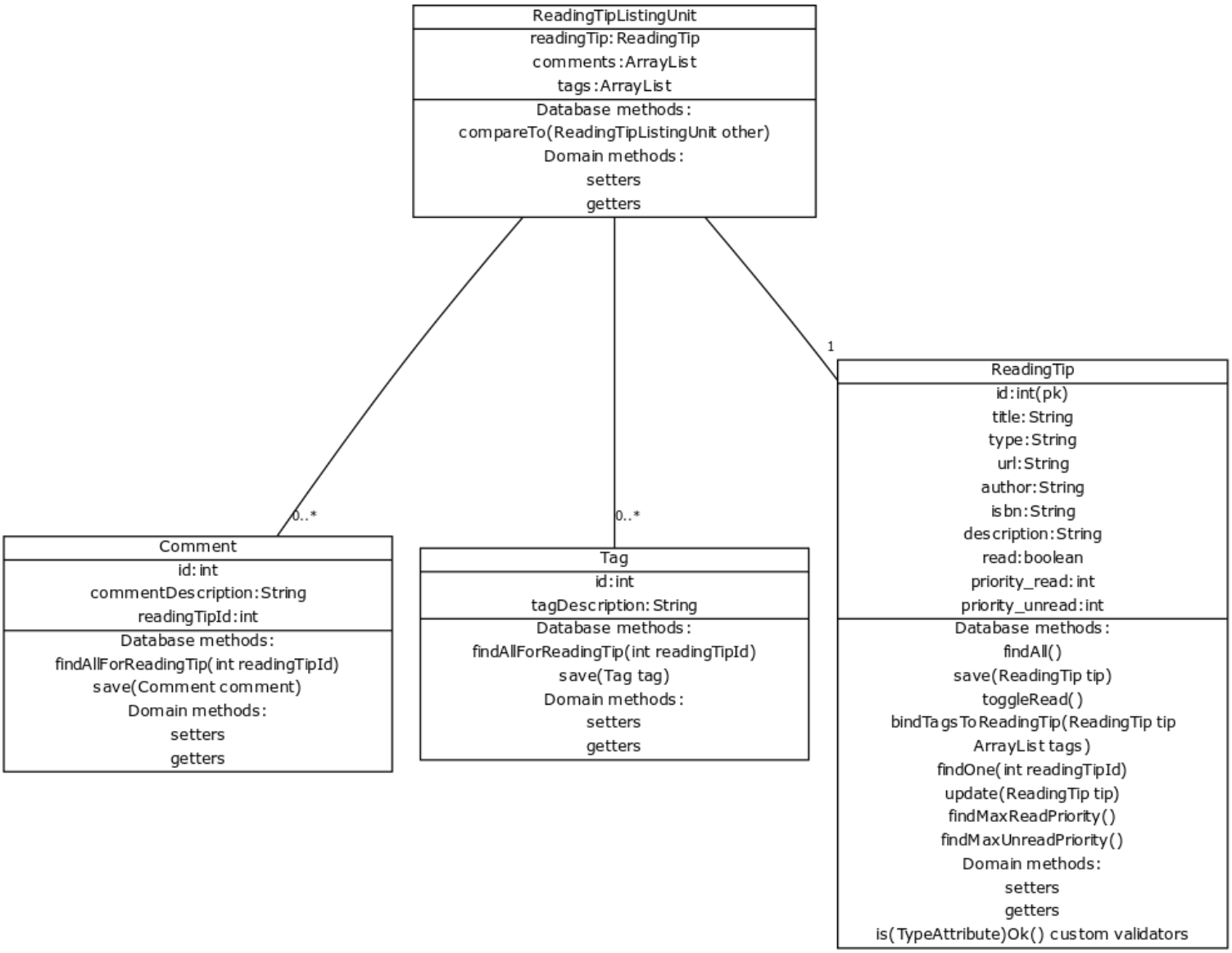
Yhdelle lukuvinkille voi antaa monta kommenttia. Kommentin attribuutista *description* löytyy itse kommentin sisältö.

Lukuvinkeille voi antaa tageja ja vinkkejä voi etsiä tagien perusteella. Lukuvinkeillä ja tageilla on näin ollen monesta-moneen suhde, jonka takia välissä on liitostaulu ReadingTipTag. Taulun Tag attribuutti *tagDescription* kuvaa tagin nimeä.

Tällä hetkellä erillistä testausympäristöä ei ole, joten testien ajaminen luo uusia olioita kehitysympäristön tietokantaan. Database-luokallamme on kuitenkin metodi helpottamaan tietokannan siivoamista testien ajamista varten.

Luokat

Luokkakaavio



Luokkien metodit ovat settereita, gettereita ja ReadingTipin tyyppeihin kuuluvien attribuuttien validaattoreita lukuun ottamatta kaikki metodeja, jotka tekevät kutsuja tietokantaan DAO-luokkien avulla. ReadingTipListingUnit sitoo ReadingTip-oliot ja siihen kuuluvat kommentit sekä tagit yhteen. Se huolehtii controllerissa ja viewissa siitä, että ReadingTipin yhteydessä voidaan näyttää myös siihen liittyvät Tag- ja Comment-oliot.

Comment-oliot ovat suoraan ReadingTip-olioihin sidottuja tietokantakaavion mukaan, kun taas Tag-oliot etsivät siihen kuuluvat ReadingTip-oliot tietokantataulun ReadingTipTags mukaan.

Luokkien attribuutit ovat suoraan verrannollisia tietokantataulun attribuutteihin. Tietokannan tyyppi text on ilmaistu koodissa tyyppinä String.

Tarkemmat metodikuvaukset löytyvät projektimme JavaDocs-kansiosta.

Metodikuvaukset (voi poistaa kun JavaDocit toteutettu)

ReadingTipListingUnit

- compareTo(ReadingTipListingUnit other)
    - Järjestää lukuvinkit lukemattomuuden perusteella (luetut ensin).

ReadingTip

- findAll()
    - Palauttaa listan kaikista ReadingTip-olioista tietokantahaun perusteella.
- save(ReadingTip tip)
    - Tallentaa parametrina annetun ReadingTip-olion tietokantaan.
- toggleRead(int readingTipId)
    - Merkkaa parametrina annetun id:n perusteella ReadingTip-olion joko luetuksi tai lukemattomaksi riippuen sen read-attribuutin tilasta.
- bindTagsToReadingTip(ReadingTip tip, ArrayList<Tag> tags)
    - Sitoo parametrina annetun listan Tag-olioita ReadingTip-olioon lisäämällä tauluun ReadingTipTag uusia rivejä saman verran kuin listassa on Tag-olioita. Jos lista on tyhjä, palataan takaisin ilman muutoksia.

Tag

- findAllForReadingTip(int readingTipId)
    - Palauttaa listan kaikista Tag-olioista jotka kuuluvat johonkin ReadingTip-olioon taulun ReadingTipTag kautta.
- save(Tag tag)
    - Tallentaa parametrina saadun Tag-olion tietokantaan. Jos samalla tagDescription-attribuutilla oleva Tag on olemassa, ei uutta tietokantariviä tehdä.

Comment

- findAllForReadingTip(int readingTipId)
    - Palauttaa listan kaikista Comment-olioista jotka kuuluvat johonkin ReadingTip-olioon attribuutin readingTipId kautta.
- save(Comment comment)
    - Tallentaa parametrina saadun Comment-olion tietokantaan. Toisin kuin Tag-olioiden save(), nämä eivät ole uniikkeja.

Bugit ja kehitysehdotukset

Selaimen uudelleenlataus

- Mikäli lukuvinkkejä on filtteröity joko hakupalkilla tai piilottamalla jo luetut lukuvinkit, ja käyttäjä merkkaa jonkun lukuvinkin luetuksi, ei sovellus muista filtterin asetuksia (tai avattuja kommenttiketjuja) kun sivu lataa uudelleen.

- Jos käyttäjä luo uuden kommentin ja lähettää sen, kaikki kommenttiketjut ovat taas suljettuina (ja filtterin asetukset taas pielessä).

Olisi hyvä, jos sovellus muistaisi tilan, jotta käyttäjä ei edes huomaisi uudelleenlatausta.

style.css refaktorointi

Tyylitiedosto style.css alkaa olla jo sekava. Tyylitiedostoa voisi refaktoroida.

ISBN tarkistus

ISBN:stä voisi tarkistaa, että se on oikeassa muodossa.