

Tehtävälista

582203 Aineopintojen harjoitustyö: Tietokantasovellus

Vesa Riekkinen

Helsingin yliopisto
Matemaattis-luonnontieteellinen tiedekunta
Tietojenkäsittelytiede

4.4.2016

1 Johdanto

Työn aihe on verkossa toimiva tehtävälista eli niin sanottu to do -lista. Sovelluksen käyttötarkoitus on, että käyttäjä voi kirjata sovellukseen tulevien päivien ja viikkojen tehtävät. Tämän tyyppisen sovelluksen tavoite on auttaa käyttäjiä heidän oman aikansa hallinnassa. Yksi perustavanlaatuisen syy tehtävälistan ylläpidolle on vähentää tehtävien järjestämisestä syntyvää aivojen kuormitusta kirjoittamalla ne esimerkiksi muistilapulle. Nykypäivänä lienee varsin luonnollista, että paperille kirjoitettujen muistilappujen lisäksi käytetään myös digitaalisia verkkopalveluita. Tämän tyyppisiä olemassa olevia verkkopalveluita ovat esimerkiksi todoist.com ja any.do.

Työn aihe on mukautettu versio valmiista aiheesta nimeltä Muistilista. Olenaisiin ero valmiiseen aiheeseen verrattuna on, että tämä sovellus järjestää tehtävät etupäässä päivämäärän mukaan. Valmis aihe Muistilista vuorostaan järjestää tehtävät etupäässä niiden prioriteetin mukaan. Tämä sovellus tulee todennäköisesti tarjoamaan myös mahdollisuuden tehtävien prioriteettien asettamiseen, mutta tämä tulee olemaan enemmänkin lisäominaisuus. Pääasiallinen tarkoitus on pitää kirjaa tehtävistä niiden suunnitellun suorituspäivämäärän mukaan.

Sovellus toteutetaan Javalla käyttäen Java EE:n web-profilia. Web-profiili tosin sisältää useita eri tekniikoita, joista tämä sovellus käyttää vain pientä osaa. Tämä sovellus käyttää nimenomaan Java Servlet ja JavaServer Pages (JSP) -tekniikoita. Näitä tekniikoita täydentämään on kehitetty myös useita erilaisia sovelluskehysjä, mutta tässä työssä ei ole tarkoitus käyttää mitään tällaista sovelluskehystä.

Servlet-tekniikan käyttö edellyttää sitä tukevan HTTP-palvelimen käyttöä. Kevyemmän mallisia servlet container -palvelimia ovat esimerkiksi Apache Tomcat ja Jetty. Nämä palvelimet ovat myös siinä mielessä erityisiä, että niitä voidaan ajaa niin sanotussa embedded-tilassa. Embedded-tila tarkoittaa olennaisesti sitä, että jokaiselle sovellukselle osoitetaan oma paikallinen HTTP-palvelin. Tällöin yksi palvelinohjelmiston kopio suorittaa vain yhtä sovellusta. Tämä on vastakohta perinteiselle mallille, jossa yksi keskitetty sovelluspalvelin pyörittää suurta joukkoa erilaisia sovelluksia. Uudentyyppistä hajautettua mallia käytetään joissakin pilvipalveluissa, jotka tyypillisesti pilkkovat kunkin palvelinkoneen fyysisesti tarjoamat resurssit useampaan virtuaaliseen ympäristöön. Tässä työssä hyödynnetään erästä tällaista pilvipalvelua.

Sovelluksen julkinen versio toimitetaan kehityksen aikana säännöllisesti Heroku-pilvipalveluun. Sovelluksen kotisivu on verkossa osoitteessa tlist.herokuapp.com. Heroku noudattaa edellä kuvattua hajautettua mallia, jossa kukin sovellus käynnistää oman palvelimensa. Palvelinohjelmiston voi siis valita vapaasti, mutta sen tulee pystyä toimimaan embedded-tilassa. Tässä työssä käytetään Apache Tomcat -sovelluspalvelinta, mutta sovellus lienee yhtä lailla yhtäso-piva Jettyn kanssa.

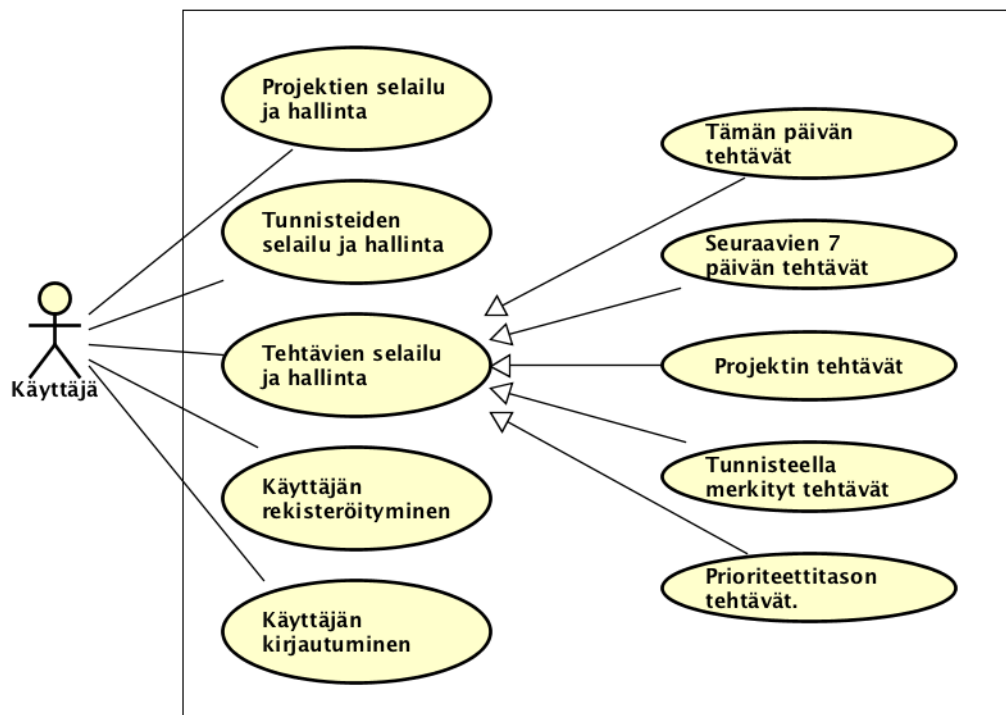
Sovelluksen tietokannaksi on valittu PostgreSQL. PostgreSQL on kurssima-teriaalissa suositeltu valinta. Heroku ei myöskään tarjoa muuta tietokantaa ilmaisversiossaan. Sovellus on tarkoitus toteuttaa SQL-standardien mukai-sella yhteensopivalla tavalla, mutta kussakin tietokannassa on tietyst erityis-piirteensä. Tällainen erityispiirre on esimerkiksi PostgreSQL:n serial-tyyppi tietokantataulujen pääavaimille. Toisin sanoen sovellus olisi todennäköisesti siirrettävissä toiseen tietokantatyyppiin kohtuullisen pienellä vaivalla, mutta sellaisenaan se toimii vain PostgreSQL-tietokannalla.

Sovelluksen käyttäjille verkossa näkyvät sivut toteutetaan HTML- ja CSS-tekniikoilla. Sivujen rakentamisessa hyödynnetään Bootstrap-kirjastoa, jo-ka sisältää suuren joukon valmiita käyttöliittymäkomponentteja. Useimmat Bootstrapin komponentit ovat siinä mielessä staattisia, että ne eivät käytä Javascript-tekniikkaa. Tässä työssä on tarkoitus hyödyntää staattisten kom-ponenttien lisäksi myös muutamaa Bootstrapin Javascript-komponenttia. Si-ten käyttäjän selaimen on tuettava myös Javascriptiä, mikä on modernien selainten perusominaisuus.

2 Yleiskuva järjestelmästä

Sovellus ryhmittelee tehtävät sillä tavalla, että kukin tehtävä kuuluu johonkin projektiin. Projektin sisällä tehtäviä voi ryhmitellä edelleen sillä tavalla, että yhdellä tehtävällä voi olla alitehtäviä. Joka tapauksessa kullakin tehtävällä on sille osoitettu suorituspäivämäärä, jota voi tarpeen mukaan myös päivit-tää. Tehtäviä voi tarkastella sekä projektikohtaisesti että kootusti. Koottu näkymä näyttää tulevan päivän ja tulevan viikon tehtävät päiväkohtaisesti.

Järjestelmä on tarkoitettu yksittäisille käyttäjille. Järjestelmä ei siis tue tii-mityöskentelyä. Tämä tarkoittaa, että kukin projekti voi kuulua vain yhdelle käyttäjälle. Toisin sanoen projektit, tunnisteet ja tehtävät ovat käyttäjäkoh-taisia. Näitä tietoja ei voi jakaa muiden käyttäjien kanssa.



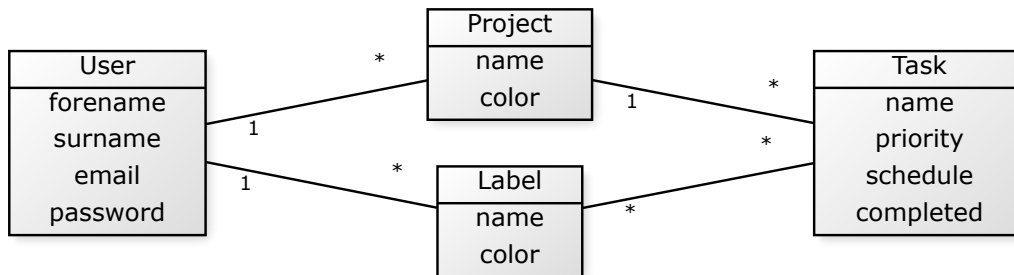
powered by Astah

Kuva 1. Käyttötapauskaavio

3 Järjestelmän tietosisältö

Järjestelmän tietosisältöä kuvaavia käsitteitä ovat käyttäjä, projekti, tunniste ja tehtävä. Järjestelmä toimii siten, että kullakin käyttäjällä on omat projektinsa, tunnisteensa ja tehtävänsä. Tässä järjestelmässä tehtävä on työn perusyksikkö, jolla on tietty päivämäärä. Tehtävät jaotellaan projekteihin. Tehtäviä voidaan luokitella myös yli projektirajojen käyttämällä tunnisteita.

Käyttäjällä voi olla mielivaltaisen määrä projekteja ja tunnisteita. Kuhunkin projektiin ja kuhunkin tunnisteeseen voi liittyä mielivaltaisen määrä tehtäviä. Projektilla ja tunnisteella on kuitenkin yksi keskeinen ero: yhdellä tehtävällä voi olla monta tunnistetta, mutta yksi tehtävä voi kuulua vain yhteen projektiin. Yksi tehtävä ei siis voi kuulua kahteen eri projektiin. Tehtävällä voi kuitenkin olla useita tunnisteita. Tehtävän tulee myös kuulua johonkin projektiin, mutta tunnisteiden merkitseminen on vapaaehtoista.

**Kuva 2.** Käsitekaavio**Taulukko 1.** Järjestelmän tietokohteiden kuvaukset

Käsite	Attribuutti	Arvojoukko	Kuvaus
User	forename	Merkkijono (max 50)	Käyttäjän etunimi
	surname	Merkkijono (max 50)	Käyttäjän sukunimi
	email	Merkkijono (max 30)	Käyttäjän sähköpostiosoite
	password	Merkkijono (max 100)	Käyttäjän salasanan tiivistelmä
Project	name	Merkkijono (max 100)	Projektin nimi
	color	Kokonaisluku	Projektille asetettu RGB-värikoodi
Label	name	Merkkijono (max 100)	Tunnisteen nimi
	color	Kokonaisluku	Tunnisteelle asetettu RGB-värikoodi
Task	name	Merkkijono (max 250)	Tehtävän nimi
	priority	Kokonaisluku	Kokonaisluku välillä 1-4.
	schedule	Päivämäärä	Suunniteltu suorituspäivä
	completed	Totuusarvo	Kertoo onko tehtävä suoritettu

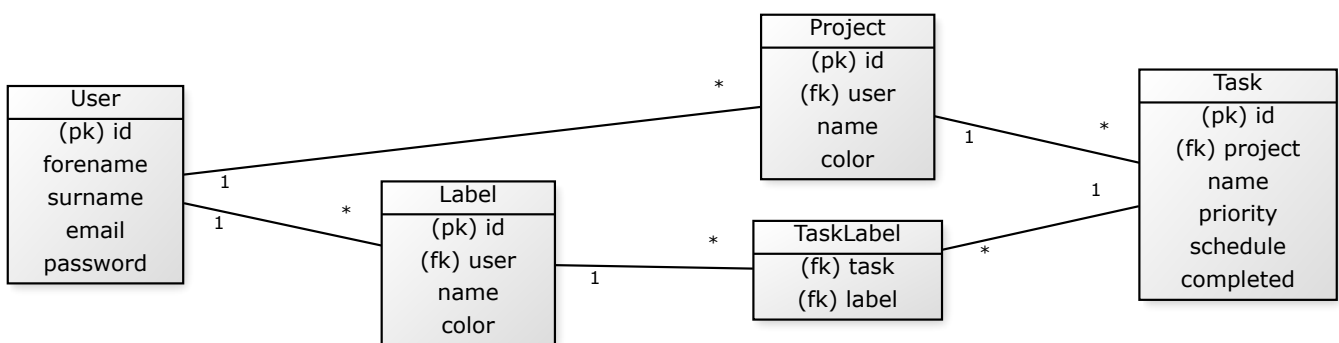
Käyttäjän salasanan tiiviste on merkkijono. Kenttä *password* sisältää merkkijonon, joka voidaan välittää salasanoja käsittelevälle metodille sellaisenaan. Merkkijono sisältää varsinaisen salasanan tiivisteen lisäksi niin sanotun suolan ja muita toteutuksen tallentamia parametreja. Merkkijonon tarkka sisältö riippuu käytettävästä salausmoduulista.

Tehtävän prioriteetti on kokonaisluku välillä 1-4. Suurinta prioriteettia merkitään luvulla 1 ja pienintä prioriteettia luvulla 3. Jos prioriteettia ei ole asetettu lainkaan, sen arvoa merkitään luvulla 4. Prioriteetti on siis sitä suurempi, mitä pienempi sen lukuarvo on, ja asteikon nollakohta on luku 4.

Käyttäjä voi valita kullekin projektille ja tunnisteelle oman värin, jota käytetään käyttöliittymässä. Värin tallennetaan RGB-muodossa. Värikomponentit tallennetaan yhtenä kokonaislukuna, mikä on yleisesti käytetty tapa. Punainen, vihreä ja sininen komponentti tallennetaan bitteihin 1-24, 8 bittiä per värikomponentti.

4 Relaatiotietokantakaavio

Tietokantakaavio voidaan johtaa käsitekaaviosta lisäämällä kuhunkin tauluun pääavain, ja esittämällä yhteydet vierasavainten avulla. Tunnisteen ja ja tehtävän välillä on monesta-moneen-tyyppinen yhteys, jonka esittämiseen tarvitaan lisäksi yksi välitaulu.



Kuva 3. Tietokantakaavio