

Tietokantasovellus - FlexDo

Henry Carlson

27. huhtikuuta 2014

Sisältö

1	Johdanto	2
1.1	Järjestelmän tarkoitus	2
1.2	Toteutus	2
2	Yleiskuva järjestelmästä	3
2.1	Käyttäjryhmät	3
2.2	Käyttötapaukset	3
3	Järjestelmän tietosisältö	5
3.1	Tietokohteet	5
4	Relaatiotietokantakaavio	7
5	Järjestelmän yleisrakenne	8
6	Käyttöliittymä ja järjestelmän komponentit	9
7	Asennustiedostot	10
8	Käynnistys-/käyttöohje	11
9	Testaus, tunnetut bugit ja puutteet, jatkokehitysideat	12
9.1	Testaus	12
9.2	Jatkokehitysideoita	12
10	Omat kokemukset	13

1. Johdanto

Tavoitteena on kehittää tehokas järjestelmä tehtävien hallintaan. Tehtävät tulisi voida priorisoida, ja lajitella omiin luokkiinsa.

1.1 Järjestelmän tarkoitus

Järjestelmän perusta on priorisoitu tehtävälista. Järjestelmän tulisi sallia käyttäjän lisätä tehtäviä listalle, asettaa tehtäville prioriteetti, ja selata niitä eri järjestyksissä (mm. prioriteetin mukaan). Lisäksi tehtävät tulisi voida luokitella erillisiin kategorioihin, mielellään hierarkisesti.

1.2 Toteutus

Järjestelmä toteutetaan Java/Tomcat verkkosovelluksena, tietokantajärjestelmänä PostgreSQL. Järjestelmän tulisi olla käytettävissä puhtaasti html muotoisena, mutta käyttöä helpottavia lisätoimintoja voidaan tarvittaessa toteuttaa javascriptillä.

2. Yleiskuva järjestelmästä

2.1 Käyttäjäryhmät

Käyttäjä

Rekisteröitynyt ja järjestelmään sisäänkirjautunut käyttäjä

2.2 Käyttötapaukset

Kirjautuminen

Rekisteröityminen

Askareen lisäys

Käyttäjä voi lisätä uuden askareen. Askareelle annetaan lisätessä nimi, mahdollinen kuvaus ja prioriteetti.

Askareen muokkaus

Käyttäjä voi muokata järjestelmässä olevan askareen prioriteettia, nimeä sekä kuvausta.

Askareen poisto

Käyttäjä voi poistaa askareen järjestelmässä. Poiston yhteydessä voidaan joko merkitä askare suoritetuksi, jolloin se jää järjestelmään erilliseen kategoriaan kirjanpitoa varten, taikka poistaa askare kokonaan järjestelmästä.

Askareiden selaus prioriteetin mukaan

Käyttäjä voi selata järjestelmässä olevia askareita prioriteettijärjestyksessä.

Luokkien lisäys ja poisto

Käyttäjä voi lisätä ja poistaa järjestelmässä olevia askareiden luokkia.

Askareiden luokittelu

Käyttäjä voi liittää askareen luokkaan, taikka poistaa sen luokasta. Yksi askare voi kuulua useaan luokkaan.

Askareiden selaus luokittain

Käyttäjä voi selata järjestelmässä olevia askareita luokan mukaan.

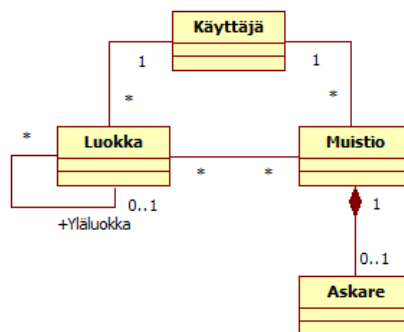
Suoritettujen askareiden selaus

Käyttäjä voi selata suoritettuja askareita suoritusa-jankohdan mukaan järjestettynä.

Luokkien hierarkkinen järjestely

Käyttäjä voi asettaa luokkia toistensa alaluokiksi, muodostaen hierarkisen luokkarakenteen.

3. Järjestelmän tietosisältö



Kuva 3.1: Käsitekaavio

3.1 Tietokohteet

Tietokohde: Käyttäjä

Attribuutti	Arvojoukko	Kuvailu
Nimi	Merkkijono	Käyttäjätunnus
Salasana	Merkkijono, pituus tiivistefunktion mukaan	Käyttäjän salasanaa vastaava tiiviste

Tietokohde: Muistio

Attribuutti	Arvojoukko	Kuvailu
Käyttäjä	Vieras avain	Linkki muistion omistavaan käyttäjään
Nimi	Merkkijono	Muistion nimi
Sisältö	Merkkijono	Muistion sisältö
Luontiaika	Aika ja päivämäärä	Muistion luomisajankohta

Muistio koostuu otsikosta ja tekstisisällöstä. Muistiosta voidaan tehdä askare linkkamalla siihen askare-tietkohde.

Tietokohde: Askare

Attribuutti	Arvojoukko	Kuvailu
Muistio	Vieras avain	Linkki muistioon, josta askareen tietosisältö löytyy
Prioriteetti	Kokonaisluku	Askareen prioriteetti
Sulkemisaika	Aika ja päivämäärä	Askareen sulkemisajankohta, jos askare on suoritettu

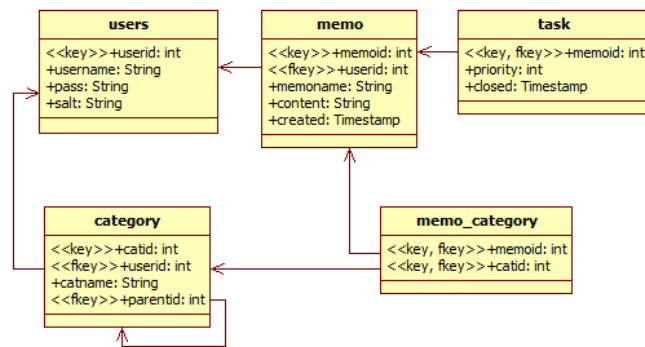
Tietokannan askare-taulu sisältää kaiken tarvittavan metatiedon jolla muistiosta voidaan tehdä askare. Jokaiseen askareeseen liittyy tasan yksi muistio.

Tietokohde: Luokka

Attribuutti	Arvojoukko	Kuvailu
Käyttäjä	Vieras avain	Linkki luokan omistavaan käyttäjään
Nimi	Merkkijono	Luokan nimi
Yläluokka	Vieras avain	Linkki mahdolliseen luokan yläluokkaan

Luokat voidaan järjestää hierarkiaan yläluokka-suhteen mukaan. Jokainen muistio voi kuulua yhteen tai useampaan luokkaan.

4. Relaatiotietokantakaavio



Kuva 4.1: Relaatiotietokantakaavio

5. Järjestelmän yleisrakenne

6. Käyttöliittymä ja järjestelmän komponentit

7. Asennustiedostot

Asenna sovellus lisäämällä Tomcat:iin sovelluspaketti (.war). Luo sen jälkeen sovellukselle 'FlexDo/META-INF/context.xml'-tiedosto kopiaimalla mallitiedosto 'FlexDo/META-INF/context.xml.dist', ja muuttamalla kohdat 'USERNAME', 'PASSWORD' ja 'DBNAME' ympäristön mukaisiksi.

8. Käynnistys-/käyttöohje

Asennuksen jälkeen sovelluksen tulisi löytyä palvelimelta aliosoitteesta '/Flex-Do/index'. Kirjaudu sisään sovellukseen hallinnointitunnuksilla (oletuksena käyttäjänimi 'admin', salasana 'admin'), ja luo tarvittavat käyttäjätunnukset.

9. Testaus, tunnetut bugit ja puutteet, jatkokehitysideat

9.1 Testaus

Sovellusta on testattu lähinnä käyttämällä sovellusta normaalisti sovelluksen kehityksen yhteydessä. Mitään varsinaista järjestelmällistä testausta (testaus-suunitelma, automatisoidut testitapaukset jne.) ei sovellukselle ole tehty.

9.2 Jatkokehitysideoita

- Askareiden haku
- Askareiden aikataulutus
- Askareiden välisiä linkkejä (wiki-tyyli)
- Askareiden välisiä vaativuuksia (Tehtävä B ei ole suoritettavissa ennenkuin tehtävä A on suoritettu)
- Projektinäköymä (kartoitettu näköymä aikataulusta ja askareiden väisistä riippuvuuksista)

10. Omat kokemukset