

DOINK

Selainpohjainen to-do application

Aineopintojen harjoitustyö: Tietokantasovellus



Jerry Mesimäki
Tommi Nikkanen

Helsingin yliopisto, Tietojenkäsittelytieteen laitos

Johdanto

Järjestelmän kuvaus: mikä on Doink?

Doink on erityisesti mobiiliselaimia varten toteutettava todo -henkinen web application. Palveluun rekisteröidytään, jonka jälkeen käyttäjä voi luoda itselleen tehtäviä, antaa niille prioriteetteja ja eräpäiviä. Tehtäviin voidaan liittää muistiinpanoja. Käyttäjä näkee vain omat ja hänelle jaetut tehtävät sekä muistiinpanot. Käyttäjä voi merkitä tehtäviä tehdyiksi, jolloin ne saavat suorituspäivän. Tehtäviä voi myös poistaa ja niiden sisältöä muokata luomisen jälkeen.

Toimintaympäristö

Doink toteutetaan mobile first -idealla, eli tarkoitus on tarjota palvelu kaikkialla siellä, missä sitä ikinä tarvitaankin. Frontend tehdään HTML5+JS -kombolla, joten palvelua pystyy käyttämään millä tahansa kyseisiä tekniikoita tukevalla päätelaitteella, ilmaiseksi.

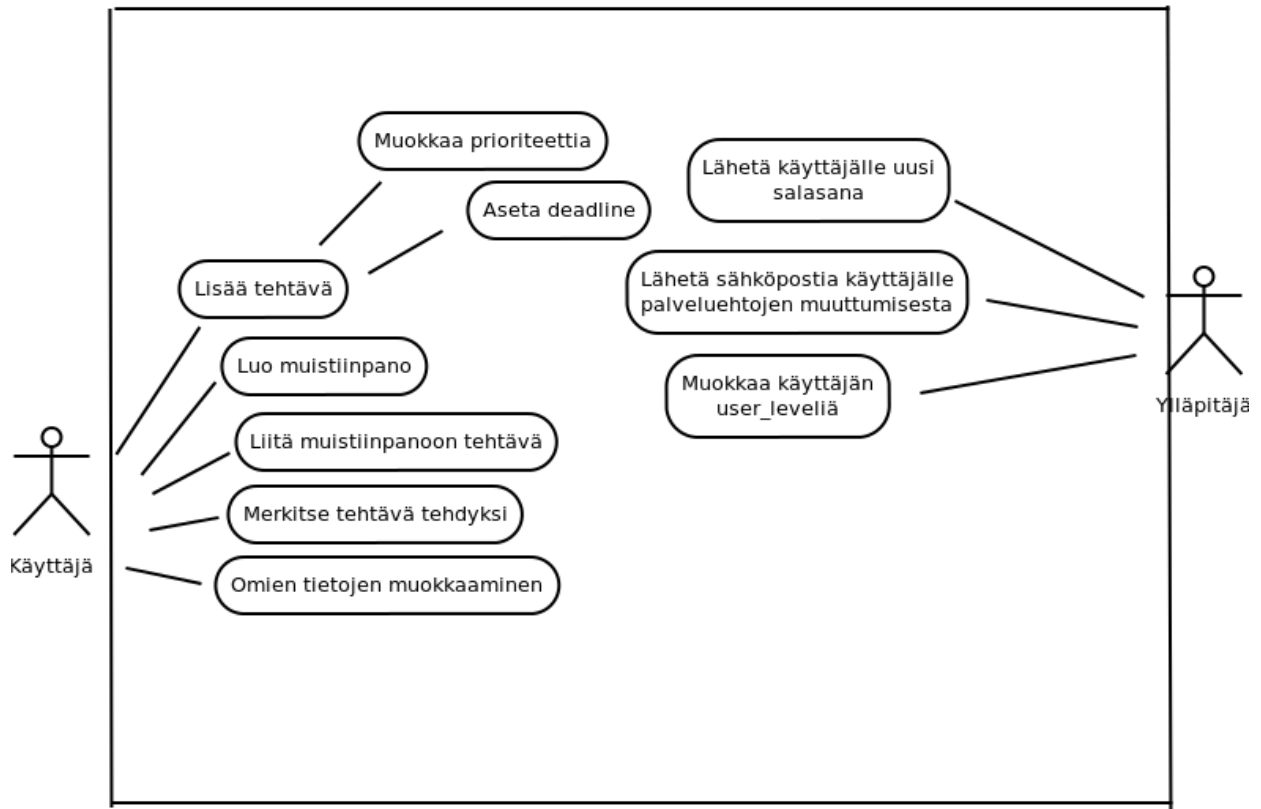
Käyttäjät ja heidän tehtävänsä tallennetaan MySQL-tietokantaan, johon syötetään dataa PHP:n avulla. Tiedon syöttö tapahtuu käyttäjän toimesta verkkosivulla. Palvelun käyttämiseen päätelaite tarvitsee modernin web-selaimen.

Jerry Mesimäki
Tommi Nikkanen

Helsingin yliopisto, Tietojenkäsittelytieteen laitos

Yleiskuva järjestelmästä

Käyttötapauskaavio



Käyttäjärühmät

Asiakas

Asiakas on rekisteröitynyt käyttäjä.

Ylläpitäjä

Palvelua ylläpitävä henkilö, joka voi generoida asiakkaille esim. uusia salasanoja unohdettujen tilalle.

Kehittäjät

Kehittäjät voivat lisätä tietokantatauluja tarpeen tullen ja muokata ulkoasua.

Käyttötapaukset

Asiakkaan käyttötapaukset

Käyttäjätunnuksen luominen

Käyttääkseen Doinkia asiakkaan tulee ensin luoda itselleen käyttäjätunnus. Luonti onnistuu sille osoitetulla sivulla, jossa asiakas syöttää sähköpostinsa, nimensä ja haluamansa salasanan. Nämä tiedot syötetään tietokantaan ja palvelun käyttö voidaan aloittaa.

Tehtävän luominen

Kirjaututtuaan palveluun asiakas voi luoda itselleen tehtäviä. Tehtävälle tulee antaa otsikko. Asiakas voi antaa tehtävälle myös valinnaisia attribuutteja kuten kuvaus, deadline ja prioriteetti.

Muistiinpanon luominen

Tehtävään voi liittää yhden tai useamman muistiinpanon. Menemällä tehtävän näkymään asiakas näkee painikkeen, jonka avulla muistiinpano luodaan. Muistiinpanoon voi syöttää haluamansa määrän tekstiä.

Kirjautuminen

Asiakkaan tulee kirjautua pystyäkseen käyttämään Doinkin tarjoamia palveluja. Kirjautuminen vaatii tunnuksen palveluun. Palveluun kirjaudutaan syöttämällä aloitussivulla sähköposti sekä salasana.

Tehtävän merkkeäminen tehdyksi

Asiakas voi halutessaan merkata tehtävän tehdyksi, jolloin se poistuu tehtävien asioiden näkymästä. Merkkeäminen tapahtuu sille osoitettua nappia painamalla. Tehtyjä tehtäviä voi tarkastella erillisestä näkymästä.

Tekemättömien tehtävien tarkastelu

Kirjaututtuaan Doinkiin käyttäjä näkee listan niistä tehtävistä, jotka hänellä ovat tekemättä. Tehtävän suhteellinen koko ruudulla riippuu sen prioriteetista ja sijainti deadlinesta. Ylimpänä on lähimmän deadline omaava tehtävä, alimpana taas kauimman. Mitä suurempi prioriteetti, sitä suuremmalta tehtävä näyttää.

Järjestelmän tietosisältö

Tehtävät

	Tietokohde: Tasks	
<u>Attribuutti</u>	<u>Arvojoukko</u>	<u>Kuvailu</u>
id	Kokonaisluku, 10 merkkiä	Tehtävän uniikki tunnisteluku
task_title	Merkkijono, 260 merkkiä	Tehtävän otsikko
task_description	Mediumtext	Tehtävän sisältö
task_complete	Kokonaisluku, 1 merkki	Onko tehtävä tehty
task_priority	Kokonaisluku, 1 merkki	Tehtävän tärkeys
task_add_date	Päivämäärä	Lisäämispäivä
task_deadline	Päivämäärä	Deadline
task_completed	Päivämäärä	Päivä jolloin tehtävä valmistui
task_owner	Merkkijono, 150 merkkiä	Tehtävän omistajan s.posti

Tehtävä, jonka asiakas on itselleen luonut. Yhden tehtävän voi omistaa kerrallaan vain yksi asiakas, mutta samanlaisia tehtäviä voi olla useilla eri asiakkailta uniikin tunnisteluvun takia.

Käyttäjät

	Tietokohde: käyttäjät	
<u>Attribuutti</u>	<u>Arvojoukko</u>	<u>Kuvaus</u>
email	varchar(150)	Sähköpostiosoite, jonka avulla käyttäjä tunnistetaan
firstname	varchar(50)	Käyttäjän etunimi
lastname	varchar(50)	Käyttäjän sukunimi
userlevel	int(1)	Käyttöoikeuksien taso
salt	varchar(250)	Salasanana salt-osio
password	varchar(250)	Salasana kryptattuna

Yksittäinen tietokantaan talletettava käyttäjä. Käyttäjä tunnistetaan sähköpostin avulla.

Muistiinpanot

	Tietokohde: muistiinpanot	
<u>Attribuutti</u>	<u>Arvojoukko</u>	<u>Kuvaus</u>
id	int(10)	Muistiinpanon tunnisteluku
task_id	int(11)	Tehtävän tunnisteluku, johon muistiinpano on liitetty
note	mediumtext	Muistiinpanon sisältö
note_edit_date	datetime	Muistiinpanon muokkauspvm
note_owner	varchar(150)	Muistiinpanon omistajan s.postiosoite

Tehtäviin voidaan liittää muistiinpanoja, jotka ovat tietokannassa omana taulunaan.

Järjestelmän php-kirjastot

Jerry Mesimäki
Tommi Nikkanen

Helsingin yliopisto, Tietojenkäsittelytieteen laitos

core.php

Pitää sisällään backendin kannalta olennaisia metodeja ja autoinclude-rutiinit.

Metodi	Parametrit	Palauttaa	Kuvaus
__autoinclude	-	-	Tekee automaattisesti includen tarvittaville luokille

cryptHandler.php

Salasanojen ja muun datan kryptaamiseen käytettävä luokka.

Metodi	Parametrit	Palauttaa	Kuvaus
getGlobalSalt	-	String salt	Palauttaa globaalin saltin jos sellainen on olemassa.
stretchHash	String pass	String hashed	Palauttaa venytetyn ja salatun salasanan.
cryptMe	String string	String	Palauttaa salatun merkkijonon

dbHandler.php

Tietokantayhteyksien hallintaan tarkoitettu luokka. Kapseloi yhteyden muodostamiseen tarvittavat tiedot - palvelimen, tietokannan ja käyttäjätiedot.

Metodi	Parametrit	Palauttaa	Kuvaus
doQuery*	String query	PDOStatement obj	Toteuttaa parametrina annetun kyselyn

loginHandler.php

Kirjautumistapahtumien käsittelyyn käytettävä luokka.

Metodi	Parametrit	Palauttaa	Kuvaus
checkLoginData	String user String pass	Boolean T/F	Tarkistaa tietokannasta onko kirjautumisen yhteydessä oleva käyttäjä/salasana -kombinaatio validi.

Jerry Mesimäki
Tommi Nikkanen

Helsingin yliopisto, Tietojenkäsittelytieteen laitos

checkPass	String pass String hash	Boolean T/F	Tarkistaa vastaako annettu salasana (pass) tietokannasta löytyvää salasanaa.
-----------	----------------------------	-------------	--

toolKit.php

Yleishyödyllinen kokoelma erilaisia apumetodeja.

Metodi	Parametrit	Palauttaa	Kuvaus
fixName	String string	String string	Muuttaa annetun merkkijonon ensimmäisen kirjaimen isoksi, ja trimmaa turhat välilyönnit ympäriltä. Käytetään nimien siistimiseen.

userCreator.php

Luokalla luodaan uusia käyttäjiä tietokantaan.

Metodi	Parametrit	Palauttaa	Kuvaus
newUser	String email String first String last String pwd	Boolean T/F	Validoi ja muotoilee annetut käyttäjätiedot, hashaa salasanan ja lisää tiedot tietokantaan.

Jerry Mesimäki
Tommi Nikkanen

Helsingin yliopisto, Tietojenkäsittelytieteen laitos

Järjestelmän komponentit

taskHandler.php

Huolehtii tasks -taulun käsittelystä tietokannassa. Luokka tarjoaa mahdollisuuden mm. luoda, muokata ja hakea käyttäjän tehtäviä valmiiden SQL-kyselyjen avulla.

loginHandler.php

Tämän avulla voidaan varmistaa, että sisään kirjautuva henkilö kirjautuu olemassa olevilla tunnuksilla, joiden sähköposti vastaa salasanaa. Salasanan käsittely tapahtuu ns. saltatussa muodossa, jolloin siihen ei pääse suoraan käsiksi edes ylläpito.

noteHandler.php

Tehtäviin liitettyjen muistiinpanojen hallitseminen SQL-kyselyillä.

dbHandler.php

Muiden luokkien käytettäväksi tarkoitettu tietokantayhteydestä huolehtiva luokka.

login.php

Näyttää käyttäjälle kirjautumissivun.

cryptHandler.php

Käyttäjän salasanan saltaaminen ja kryptaus ennen tietokantaan siirtoa.

AJAXHandler.php

Käyttäjän selaimessa tekemien toimintojen välittäminen tietokantaa muokkaavien luokkien käsiteltäväksi.

core.php

Luokkien autolodaaminen tarvittaessa.

userCreator.php

Uusien käyttäjien syöttäminen tietokantaan.

Käyttöliittymä

Jerry Mesimäki
Tommi Nikkanen

Helsingin yliopisto, Tietojenkäsittelytieteen laitos

Asennustiedot

Doink voidaan asentaa palvelimelle esimerkiksi [MAMP](#) -ohjelmiston avulla, josta löytyy tuki Apacheille, MySQL:lle sekä PHP:lle, joita Doink käyttää. Yksinkertaisimmillaan asennus tapahtuu siirtämällä tiedostot palvelimelle ja muokkaamalla dbHandler.php:ssä sijaitsevia kredentiaaleja omaa ympäristöä vastaaviksi.

Palvelimelle tulee ennen ohjelmiston käyttöä luoda tietokanta, jossa on kolme taulua: Users, Tasks, Notes.

tasks

```
CREATE TABLE IF NOT EXISTS `tasks` (  
  `id` int(10) NOT NULL AUTO_INCREMENT,  
  `task_title` varchar(260) NOT NULL,  
  `task_description` mediumtext,  
  `task_complete` tinyint(1) NOT NULL DEFAULT '0',  
  `task_priority` int(1) NOT NULL DEFAULT '1',  
  `task_add_date` datetime NOT NULL,  
  `task_deadline` datetime DEFAULT NULL,  
  `task_completed` datetime DEFAULT NULL,  
  `task_owner` varchar(150) NOT NULL,  
  PRIMARY KEY (`id`)  
)
```

users

```
CREATE TABLE IF NOT EXISTS `users` (  
  `email` varchar(150) NOT NULL,  
  `firstname` varchar(50) DEFAULT NULL,  
  `lastname` varchar(50) DEFAULT NULL,  
  `userlevel` int(1) NOT NULL,  
  `salt` varchar(250) DEFAULT NULL,  
  `password` varchar(250) NOT NULL,  
  PRIMARY KEY (`email`)  
)
```

notes

```
CREATE TABLE IF NOT EXISTS `notes` (  
  `id` int(10) NOT NULL AUTO_INCREMENT,  
  `task_id` int(11) NOT NULL,  
  `note` mediumtext COLLATE utf8_bin NOT NULL,  
  `note_edit_date` datetime NOT NULL,  
  `note_owner` varchar(150) COLLATE utf8_bin NOT NULL,  
  PRIMARY KEY (`id`)  
)
```

Jerry Mesimäki
Tommi Nikkanen

Helsingin yliopisto, Tietojenkäsittelytieteen laitos

Käynnistys- / Käyttöohje

Jerry Mesimäki
Tommi Nikkanen

Helsingin yliopisto, Tietojenkäsittelytieteen laitos