**TODOAPP**

**Dokumentaatio v.0.1**

# Johdanto

TodoApp on monipuolinen, kollaboratiivinen tehtävälista. Sovelluksen avulla käyttäjä pystyy hallitsemaan helposti työn ja vapaa-ajan tehtäviään yksin ja yhdessä muiden käyttäjien kanssa.

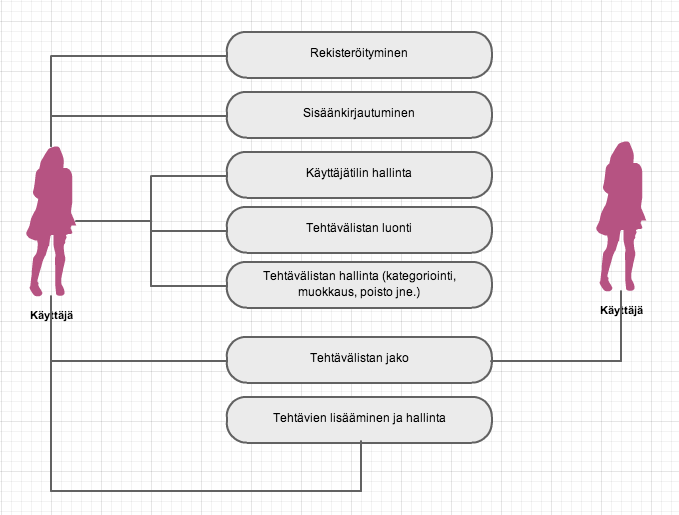
Sovellus toteutetaan käyttäen node.js-alustaa [http://nodejs.org/], joka on kevyt, tapahtumavetoinen palvelinpään ohjelmointiin tarkoitettu avoimen lähdekoodin projekti. Pysyväisdatan tallennukseen käytetään MySQL-tietokantaa.

Sovellusta voidaan ajaa tietojenkäsittelytieteen laitoksen koneissa, joihin on asennettu node.js tai vaihtoehtoisesti Heroku:n [http://heroku.com/] pilvipalvelussa.

Sovellus tukee ensimmäisessä vaiheessa ainoastaan Google Chrome –selainta versiosta 23 alkaen. Selaimen tulee sallia JavaScriptin suorittaminen, sillä käyttöliittymä on dynaaminen ja tosiaikainen.

# Yleiskuva järjestelmästä

Järjestelmässä on vain yhden kaltaisia käyttäjiä. Sovelluksen käyttö edellyttää rekisteröitymistä ja sisäänkirjautumista.



# Käyttötapaukset

## Käyttäjät

## Rekisteröinti

Voidakseen käyttää sovellusta, käyttäjän tulee rekisteröityä. Rekisteröityminen tapahtuu täyttämällä sähköpostiosoite ja salasana rekisteröitymislomakkeeseen.

## Sisäänkirjautuminen

Käyttäjä voi kirjautua sovellukseen syöttämällä käyttäjätunnuksensa ja salasanansa sisäänkirjautumislomakkeeseen.

## Uloskirjautuminen

Sisäänkirjautunut käyttäjä voi lopettaa sovelluksen käytön kirjautumalla ulos.

## Käyttäjätietojen muokkaus

Käyttäjä voi vaihtaa salasanaansa ja kirjautumiseen käytettävää sähköpostiosoitetta.

## Tehtävälista

## Tehtävälistan luonti

Sisäänkirjautunut käyttäjä voi luoda uuden tehtävälistan.

## Tehtävälistan muokkaus

Sisäänkirjautunut käyttäjä voi muokata luomansa tehtävälistan tietoja (kategoriat ym.).

## Tehtävälistan poisto

Sisäänkirjautunut käyttäjä voi poistaa luomansa tehtävälistan ja kaiken sen sisältämän tiedon.

## Tehtävälistan jako

Sisäänkirjautunut käyttäjä voi jakaa luomansa tehtävälistan valitsemilleen käyttäjille. Lisätyillä käyttäjillä on oikeutta lisätä, poistaa ja muokata tehtävälistaan kuuluvia tehtäviä, mutta heillä ei ole mahdollisuutta muokata itse tehtävälistaan esimerkiksi lisäämällä muita käyttäjiä tai vaihtamalla sen nimeä.

## Tehtävät

## Tehtävän lisääminen

Sisäänkirjautunut käyttäjä voi lisätä uusia tehtäviä hallitsemiinsa tehtävälistoihin.

## Tehtävän muokkaaminen

Sisäänkirjautunut käyttäjä voi muokata hallitsemiensa tehtävälistojen tehtäviä.

## Tehtävän poistaminen

Sisäänkirjautunut käyttäjä voi poistaa tehtäviä hallitsemistaan tehtävälistoista.

## Kategoriat

## Kategorian lisääminen

Sisäänkirjautunut käyttäjä voi lisätä yhden kategorian luomiinsa tehtävälistoihin. Kategorioilla on hierarkkinen rakenne, jolloin lisättävä kategoria voi olla jonkin toisen kategorian alikategoria esim. Työ -> Laskut.

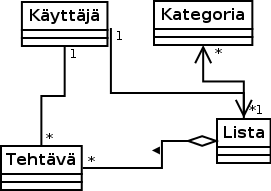
## Kategorian muokkaaminen

Sisäänkirjautunut käyttäjä voi muokata luomiensa kategorioiden nimiä.

## Kategorian poistaminen

Sisäänkirjautunut käyttäjä voi poistaa kategorian luomistaan tehtävälistoista.

# Järjestelmän tietosisältö



**Tietokohde: Käyttäjä**

|  |  |  |
| --- | --- | --- |
| **Attribuutti** | **Arvojoukko** | **Kuvailu** |
| Tunnus. | Kokonaisluku. | Uniikki tunnus, joka identifioi käyttäjän. |
| Sähköposti | Merkkijono, max. 80 merkkiä. | Käyttäjän sähköpostiosoite, joka toimii käyttäjätunnuksena. |
| Salasana | Merkkijono, max 255 merkkiä. | Käyttäjän salasanasta muodostettu turvallinen tiiviste. |
| Etunimi | Merkkijono, max. 100 merkkiä | Käyttäjän etunimi. |
| Sukunimi | Merkkijono, max. 100 merkkiä | Käyttäjän sukunimi. |

Käyttäjä on järjestelmään rekisteröitynyt käyttäjä. Käyttäjätunnuksena käytetään käyttäjän sähköpostiosoitetta. Samalla sähköpostiosoitteella voi rekisteröidä vain yhden käyttäjän. Käyttäjän tulee olla kirjautunut sisään luodakseen listoja, tehtäviä ja kategorioita.

**Tietokohde: Lista**

|  |  |  |
| --- | --- | --- |
| **Attribuutti** | **Arvojoukko** | **Kuvailu** |
| Tunnus. | Kokonaisluku. | Uniikki tunnus, joka identifioi listan. |
| Nimi | Merkkijono, max. 255 merkkiä. | Listalle annettava nimi. |
| Kategoria | Kokonaisluku. | Listan kategoria. Arvo on kategorian uniikki tunnus. |
| Luomishetki | Aikaleima. | Serverin aikaleima ajalta, jolloin tehtävä on luotu. |
| Muokkaushetki | Aikaleima. | Serverin aikaleima ajalta, jolloin tehtävää on muokattu. |

Lista on tehtävälistan perusyksikkö. Kullakin listalla on yksi kategoria.

**Tietokohde: Kategoria**

|  |  |  |
| --- | --- | --- |
| **Attribuutti** | **Arvojoukko** | **Kuvailu** |
| Tunnus. | Kokonaisluku. | Uniikki tunnus, joka identifioi kategorian. |
| Yläkategoria. | Kokonaisluku tai tyhjä. | Yläkategorian tunnus, jos kategoria on alikategoria. |
| Luoja | Kokonaisluku. | Kategorian luoneen käyttäjän tunnus. |
| Nimi | Merkkijono, max 255 merkkiä. | Kategorialle annettava nimi. |

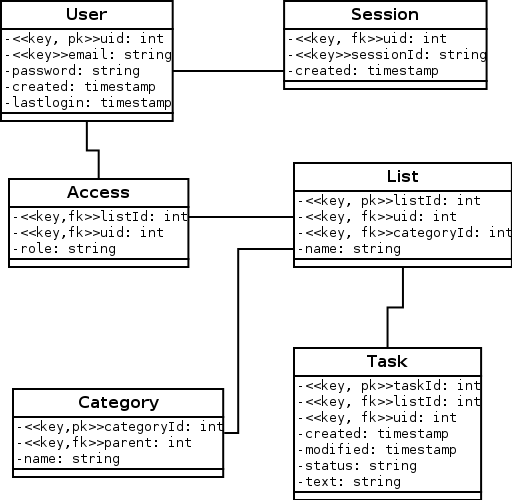
Kategoria määrittelee tehtävälistan luokittelun. Kategoria voi olla ylä- tai alatason kategoria. Samannimisiä kategorioita voi olla useampia. Kategorioiden luominen ja hallinta on käyttäjän vastuulla.

**Tietokohde: Tehtävä**

|  |  |  |
| --- | --- | --- |
| **Attribuutti** | **Arvojoukko** | **Kuvailu** |
| Tunnus | Kokonaisluku. | Uniikki tunnus, joka identifioi tehtävän. |
| Lista | Kokonaisluku. | Viittaus listan tunnukseen, johon tehtävä kuuluu. |
| Luoja | Kokonaisluku. | Viittaus käyttäjään, joka on luonut tehtävän alunperin. |
| Muokkaaja | Kokonaisluku. | Viittaus käyttäjään, joka on muokannut tehtävää viimeksi. |
| Luomishetki | Aikaleima. | Serverin aikaleima ajalta, jolloin tehtävä on luotu. |
| Muokkaushetki | Aikaleima. | Serverin aikaleima ajalta, jolloin tehtävää on muokattu. |
| Tila | Merkkijono, max 50 merkkiä. | Tehtävän tila, ”kesken” tai ”valmis”. |
| Teksti | Merkkijono, max 255 merkkiä. | Tehtävän tekstiosio, jossa määritellään mitä tehtävä koskee. |

Tehtävä kuuluvat aina johonkin listaan ja niillä on jokin selite. Tehtävällä on useita eri tiloja; tehtävä voi olla kesken tai valmis.

# Relaatiotietokantakaavio



Relaatiotietokantakaavio poikkeaa käsitemallista siten, että siihen on lisätty pari aputaulua [Access ja Session]. Näistä Session taulussa seurataan sisäänkirjautuneiden käyttäjien sessioita. Access taulussa puolestaan mallinnetaan käyttäjien pääsynhallintaa eri listoihin.

# Järjestelmän yleisrakenne

Järjestelmä noudattaa MVC-arkkitehtuurimallia. Serveripuolella mallit sijaitsevat models-kansiossa, kontrollerit routes-kansiossa ja näkymät views-kansiossa. Järjestelmän asetukset löytyvät config-kansiosta – osa asetuksista annetaan ympäristömuuttujissa (tarkemmin myöh.). Public-kansiossa sijaitsee kuvat, tyylitiedostot sekä javascript-tiedostot ja –templatet. Css-tiedostot generoidaan less-kansiossa olevien less-tyyilitiedostojen pohjalta.

Serveripuolelta tarjoillaan ainoastaan kaksi näkymää, sivu sisäänkirjautumista ja rekisteröitymistä varten sekä erillinen sivu sisäänkirjautuneille käyttäjille. Varsinaiset näkymät rakennetaan public-kansion javascript-tiedostojen ja –templatejen avulla.

Kansio node\_modules sisältää node.js:n käyttämät kirjastot.

├── config

├── less

│   └── includes

├── model

├── node\_modules

├── public

│   ├── css

│   ├── img

│   ├── js

│   │   ├── controllers

│   │   └── lib

│   └── templates

├── routes

└── views

# Järjestelmän komponentit

# Käyttöliittymä

# Asennustiedot

Sovellus on asennettu ajettavaksi Herokussa. Tämänhetkinen url on dome.herokuapp.com.

Sovelluksen lähdekoodin saa kloonattu GitHubista komennolla

***git clone https://github.com/JaakkoL/todoapp.git***.

Sovelluksen saa asennettua muuhun ympäristöön muokkaamalla config-kansiossa olevaa production.js-tiedostoa. Sovellukselle tulee antaa pääsy MySQL-tietokantaan antamalla tunnukset tiedostossa seuraavaan tyyliin:

***var config = {***

***db : {***

***host : 'example.com',***

***user : 'username,***

***password : 'password',***

***database : 'databsename'***

***}***

***}***

Lisäksi sovellus tarvitsee node (http://nodejs.org/) ja npm (npmjs.org) ohjelmat. Npm tulee tätä nykyä tuoreen node.js-asennuksen mukana.

Ohjelman riippuvuudet sijaitsevat ***package.json***-tiedostossa ja ne saa asennettua ajamalla ohjelman juuressa komento:

***npm install***.

# Käynnistys- /käyttöohje

Sovelluksen saa käynnistettyä, kun riippuvuudet on ladattu ja tietokanta konfiguroitu. Käynnistyskomento sovelluksen juuressa on:

***NODE\_ENV=<konfiguraationympäristö> node app.js***

Ohjelma pyörii oletusarvoisesti portissa ***8080***, mutta portin voi syöttää myös ympäristömuuttujassa seuraavalla tavalla:

***PORT=XX NODE\_ENV=<konfiguraationympäristö> node app.js***

# Testaus, tunnetut bugit ja putteet & jatkokehitysideat

# Omat kokemukset

# Liitteet