

To do -list sovellus

1. Johdanto:

1.1 Kuvaus

Muistilista, toiselta nimeltään To do -list, on palvelu, jossa voi tallettaa omia tärkeitä muistettavia tehtäviä ja askareita. Palvelun tarkoituksena on antaa käyttäjälle toiminnallinen sovellus, jossa voi suunnitella päivän tehtävät ja luokitella niitä tärkeysjärjestykseen ja asettaa deadline-ajat, johon mennessä tietty tehtävä pitäisi olla tehty. Palvelu valmistuessaan tarjoaa käyttäjälle mahdollisuuden tehostaa omaa ajankäyttöä ja parantaa produktiivisuutta unohtamatta eri tehtävien tärkeitä päivämääriä.

1.2 Toteutus- /toimintaympäristö

Sovellus toimii valmistuessaan `users.cs.helsinki.fi` palvelimella.

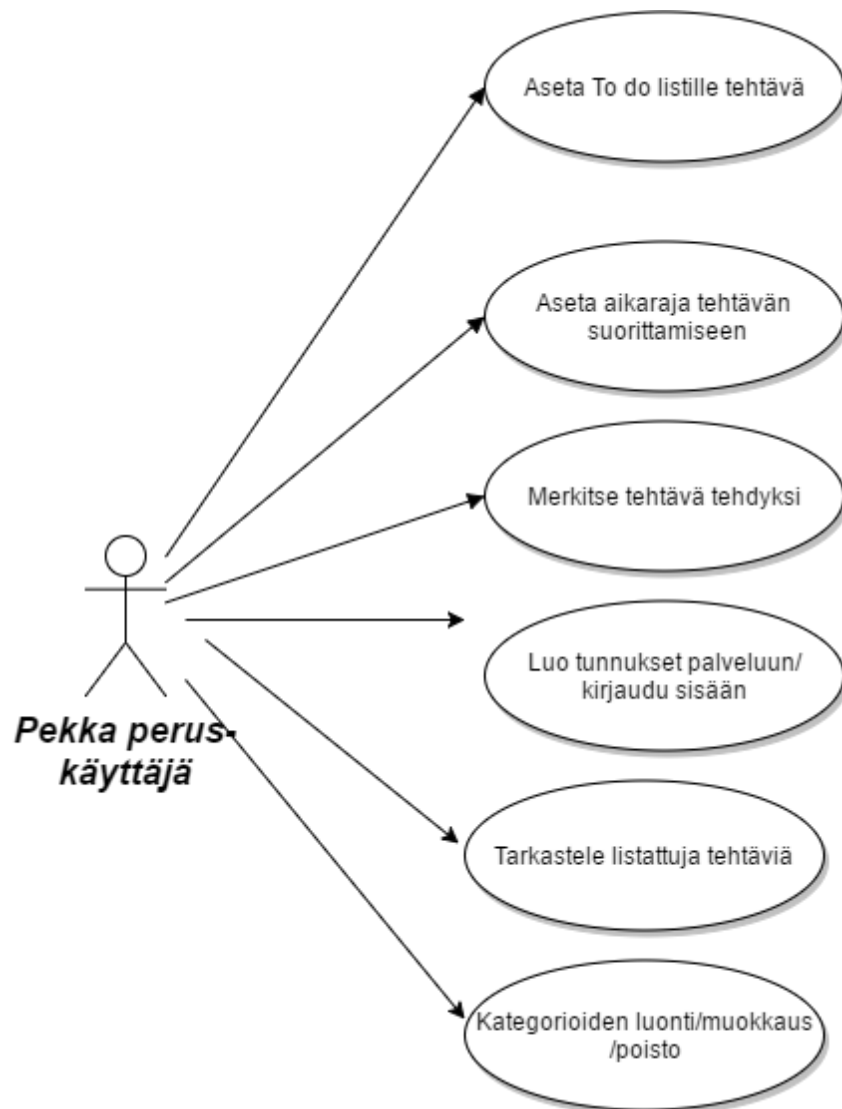
1.3 Sovelluksen toimintavaatimukset

Sovellus toteutetaan PHP:llä ja näin ollen web-sovelluksen alustajärjestelmä vaatii PHP-tuen. Lisäksi selaimelle asettuneen sovelluksen rakentuessa vaatimus selaimen Javascript-tuelle.

Tietokannan hallintajärjestelmä hoidetaan PostgreSQL:llä.

2. Yleiskuva järjestelmästä

2.1 Käyttötapauskaavio



2.2 Käyttäjärühmät

Sovelluksella on kahden tyyppisiä käyttäjiä, eli peruskäyttäjä Pekka, joka käyttää sovellusta arjen hallintaan ja ylläpitäjä, joka sen lisäksi, että käyttää sovellusta omana muistilistanaan, vastaa myös sovelluksen toiminnasta ja hallinnollisista toiminnoista, kuten käyttäjätilien hallinta yms.

2.3 Käyttötapauskuvaukset

Pekka peruskäyttäjä:

Aseta Todo listalle tehtävä

listalle laitetaan lomakkeella tehtävä, joka vaatii tekemistä

Tehtävän deadlinen asettaminen

lomakkeella asetetaan päivämäärä suoritettavalle tehtävälle

Tehtävän asettaminen tehdyksi

Tehtävää muokkaamalla voi ruksia tehtävän tehdyksi

Tunnusten luonti/kirjautuminen

Käyttäjä luo tunnukset lomakkeella ja kirjautuu palveluun sisään

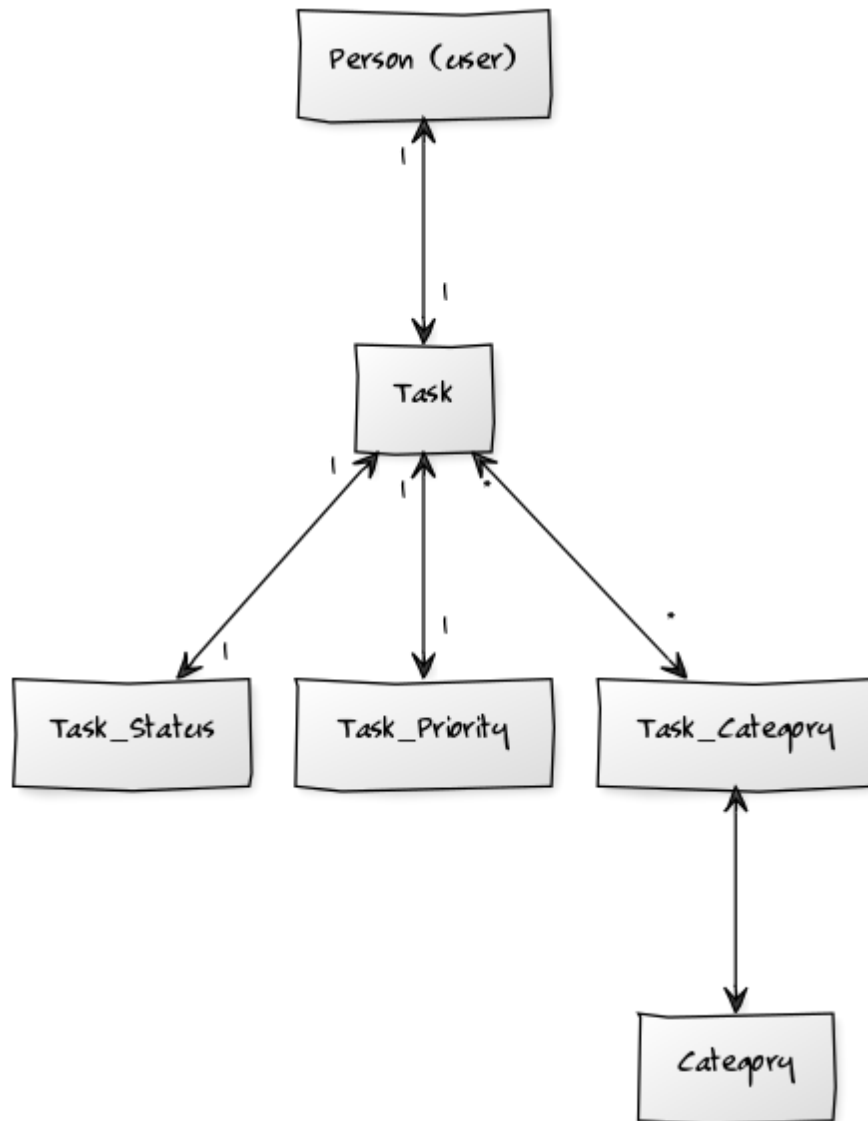
Todo - listan tarkastelu

Käyttäjä voi tarkastella listaamiaan tehtäviä

Kategorioiden luonti/muokkaus/poisto

Käyttäjä voi luokitella tehtävänsä kategorioihin aiheen/kiireellisyyden mukaan

3. Järjestelmän tietosisältö



Tietokohde: Person

Attribuutti	Arvojoukko	Kuvailu
Id	SERIAL	uniikki id käyttäjälle
username	VARCHAR	käyttäjän nimi
password	VARCHAR	käyttäjän salasana

Käyttäjä kirjautuessaan tekee tunnukset sivulle, eli käyttäjänimi ja salasana ja käyttäjät tunnistetaan yksilöllisen id:n avulla. Kullakin käyttäjällä on käytössään yksi tehtävälista jossa tehtävät on.

Tietokohde: Task

Attribuutti	Arvojoukko	Kuvailu
id	SERIAL	tehtävän id
taskname	VARCHAR	tehtävän nimi
info	VARCHAR	tehtävän kuvaus
deadline	DATE	tehtävän aikaraja
added	DATE	tehtävän lisäyspäivä
status_id	INTEGER FK	tehtävän status
priority_id	INTEGER FK	tehtävän tärkeysaste

Tehtävä on käyttäjän lisäämä tehtävä jolla on nimi, ja tarkempi kuvaus, sekä käyttäjä asettaa tehtävän luodessaan sille deadline päivämäärän mihin mennessä tehtävä tulisi olla tehty. Käyttäjä voi myöhemmin muokata tehtävää ja joko siirtää deadlinea tai muuttaa tehtävän tilan tehdyksi (oletusarvoisesti tehtävä on aina tekemätön).

Tehtävään voi asettaa yhden tai useamman kategorian, jolla määritellä tehtävän sisältöä sekä kiireellisyysasteen, jolla tehtäviä voi sitten listata eri tavalla ja tehtävät näkyvät eri värillä.

Tietokohde: Category

Attribuutti	Arvojoukko	Kuvailu
id	SERIAL	kategorian yksilöivä id
category_name	VARCHAR	kategorian nimi

Käyttäjän luoma kategoria, sama kategoria voi liittyä moneen tehtävään.

Tietokohde: Task_priority

Attribuutti	Arvojoukko	Kuvailu
id	SERIAL	id
Priority_name	VARCHAR	1 = High priority, 2 = Medium ja 3 = Low
Priority_class	INTEGER	1-3

Kullekin tehtävälle voidaan asettaa kiireellisyysaste. Yhdellä tehtävällä voi olla vain yksi luokitus kerrallaan ja monella eri tehtävällä voi olla sama kiireellisyysluokitus.

Tietokohde: Task_status

Attribuutti	Arvojoukko	Kuvailu
id	SERIAL	id
Priority_name	VARCHAR	1 = Not started, 2 = In progress ja 3 = Completed
Priority_class	INTEGER	1-3

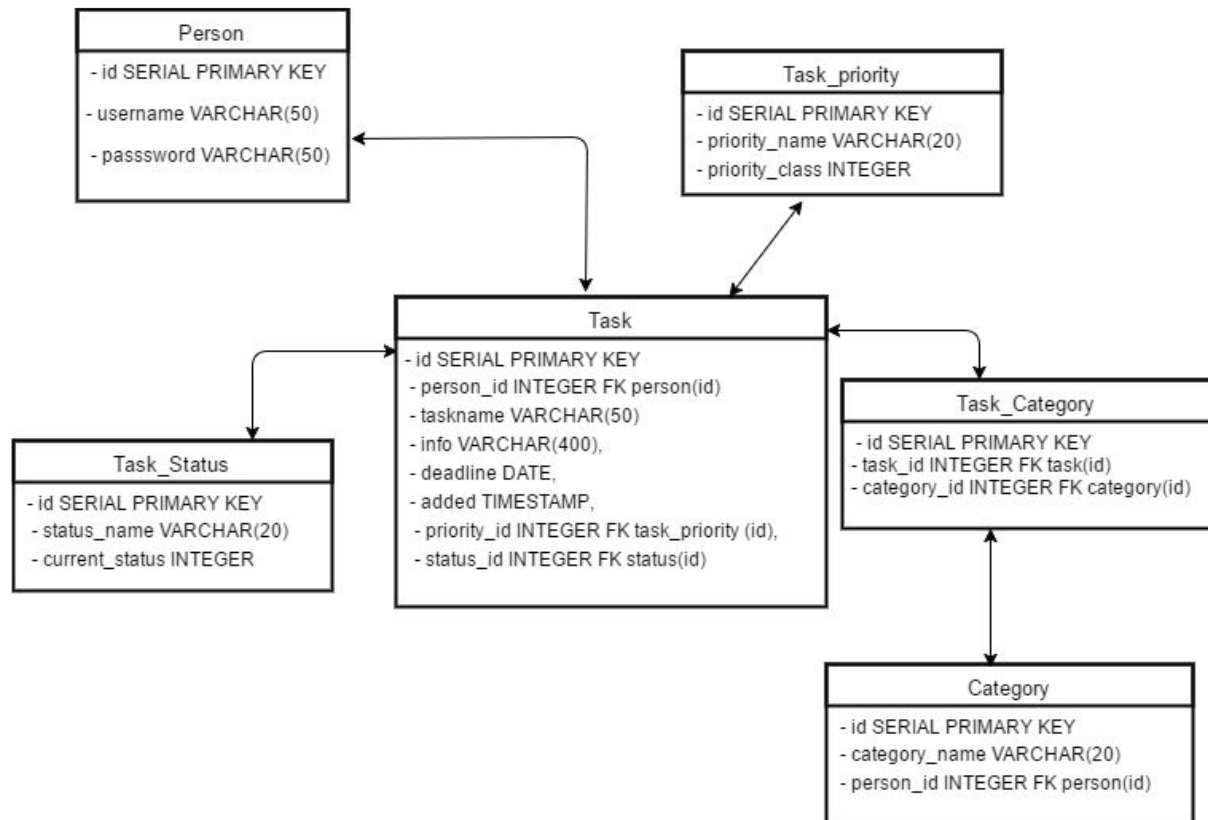
Kullekin tehtävälle voidaan asettaa tila. Yhdellä tehtävällä voi olla vain yksi status kerrallaan ja monella eri tehtävällä voi olla sama status, esim. tehtävä tehty aka. completed.

Tietokohde: Task_category

Attribuutti	Arvojoukko	Kuvailu
id	SERIAL	id
category_id	INTEGER FK	kategorian id
task_id	INTEGER FK	tehtävän id

Liitostaulu Categoryn ja Taskin välillä. Kukin tehtävä voidaan laittaa yhteen tai useampaan kategoriaan. Yhdellä tehtävällä voi olla useampi kategoria johon kuuluu ja monella eri tehtävällä voi olla sama kategoria..

4. Relaatiotietokantakaavio

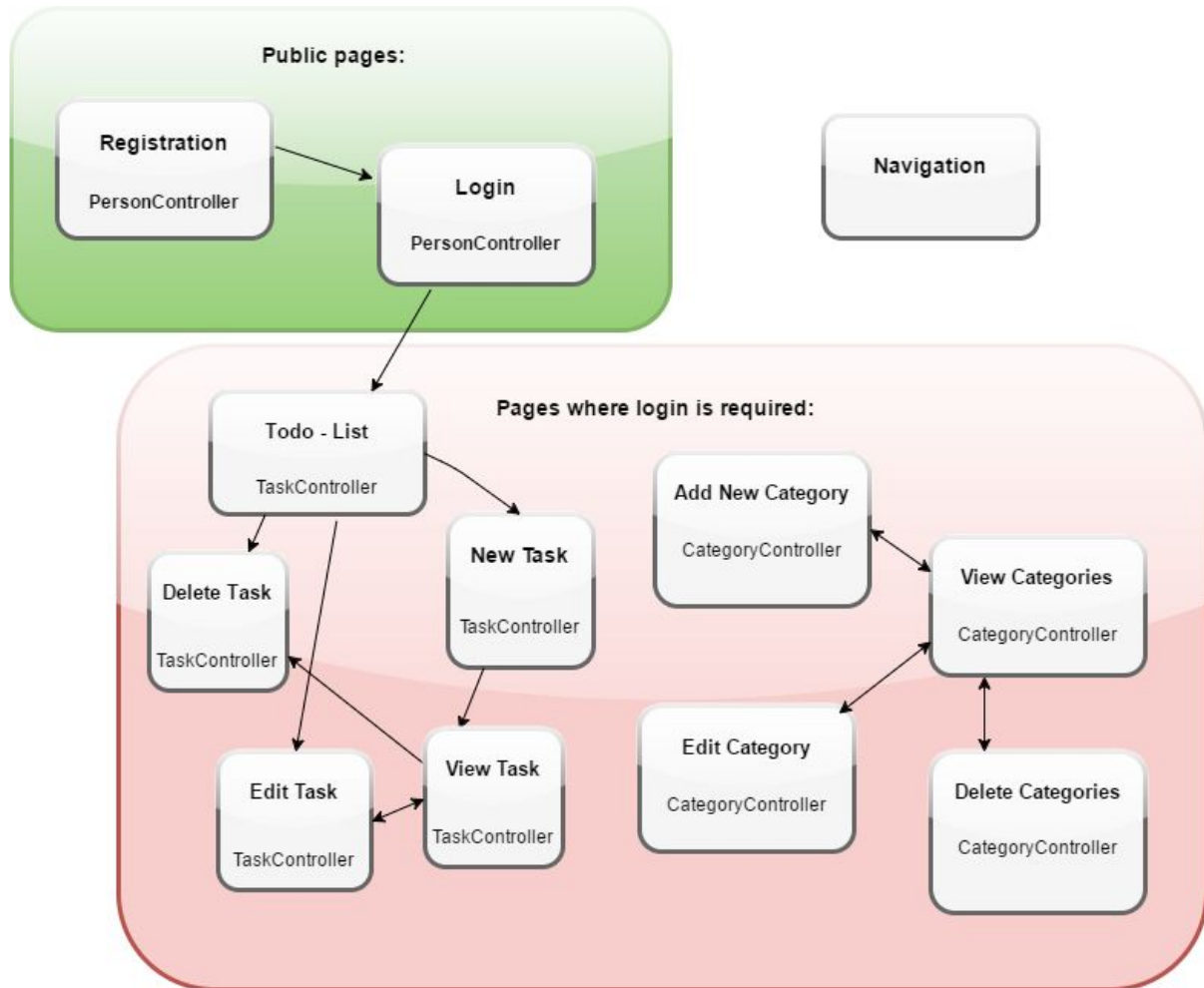


5. Järjestelmän yleisrakenne

Tietokantasovellusta tehdessä on noudatettu MVC-mallia. Kontrollerit, näkymät ja mallit sijaitsevat app-hakemiston alihakemistoissa nimeltä controllers, views- ja models. Lib -kansioista löytyvät apukirjastot, joiden toimintaa kontrollerit, näkymät ja mallit perivät. Assets - hakemistossa on CSS- ja JavaScript-tiedostot. Projektin dokumentaatio sijaitsee doc-kansiossa ja SQL-lausekkeet (create, testdata ja delete) ovat nimensä mukaisessa kansiossa.

Sovellus tallentaa tiedot kirjautuneesta käyttäjästä Sessioniin. Tiedostojen nimeäminen on pyritty hoitamaan metodeja ja luokkia kuvaavasti. Tiedostonimet ovat kirjoitettu pienellä ja _ toimii välilyönnin korvikkeena.

6. Käyttöliittymä ja järjestelmän komponentit



7. Asennustiedot

Sovelluksen voi asentaa toiselle palvelimelle kopioimalla sovellustiedostot hakemistoon, jolla on yhteys palvelimelta nettiin. Nykyisessä projektissa tätä hakemistoa vastaan usersin htdocs-hakemisto. Sen jälkeen tulisi päivittää libs.config.php tiedoston tietokannan yhteystiedot ja esimerkiksi libs/config.php.dist tarjoaa esimerkin asetustiedoston malliksi. Tietokantataulut voidaan luoda ajamalla create_tables.sql ja saamaan tietokanta pystyyn.

8. Käynnistys- / käyttöohje

Osoite <http://nooramar.users.cs.helsinki.fi/tsoha/>

Tunnukset (tunnus / salasana)

user / user

Noora / noora