

# **TIETOKANTASOVELLUS: PALKANLASKENTA JA TUNTIKIRJANPITO**

## **SISÄLLYSLUETTELO**

- 1. JOHDANTO**
- 2. YLEISKUVA JÄRJESTELMÄSTÄ**
  - 2.1. KUVAUS**
  - 2.2. KÄYTTÖTAPAUKSET**
    - 2.2.1 KÄYTTÖTAPAUUSKAAVIO**
    - 2.2.2 KÄYTTÄJÄT**
    - 2.2.3 KÄYTTÖTAPAUSKUVAUKSET**
- 3. TIETOSISÄLTÖs**
- 4. RELAATIOKAAVIO**
- 5. JÄRJESTELMÄN YLEISRAKENNE (ei valmis)**
- 6. KÄYTTÖLIITTYMÄ**
- 7. ASENNUSTIEDOT**
- 8. KÄYTTÖOHJE**
- 9. TESTAUS, BUGIT (ei valmis)**
- 10. OMAT KOKEMUKSET (ei valmis)**
- 11. LIITTEET (ei valmis)**

## 1. JOHDANTO

Tavoitteena on rakentaa kevyt palkanlaskenta ja tuntikirjanpito ohjelmisto pk-yrityksille. Järjestelmä tulee yrityksen sisäiseksi työkaluksi, joten sen käyttäjiä ovat tietenkin kaikki yrityksen työntekijöitä, jotka kuitenkin tässä tapauksessa jaotellaan johtajiin ja työntekijöihin – mahdollisesti jopa työntekijöihin, vuoropäälliköihin ja johtoryhmään, joista jälkimmäiselle tuotetaan selkeä yleisraportti. Järjestelmä helpottaa ja nopeuttaa pienten yritysten työntekijöiden työmäärien ja palkkojen seurantaa, sekä helpottaa näiden analysoimista.

Projektin aion toteuttaa perinteisen LAMP stackin päälle eli tässä tapauksessa Linux, Apache, MySQL ja PHP. Tämä pystytetään Linux virtuaalipalvelimelle (UpCloud) siten, että tietokantapalvelin ja itse web-palvelin ovat eriytetty – harjoittelen siis samalla vikasietoisia web-sovellus arkkitehtuureja. Projektin tietokantapalvelin on käytettävissä julkiverkon puolelta, joten pystyn helposti kehittämään sovellusta koneella kuin koneella lokaalisti, kunhan konfiguroin tietokannan tunnukset manuaalisesti.

Projektin palvelimet ovat jo pystyssä ja toiminnassa – luonnollisesti salasanoilla suojattuina:

- DB: (185.20.136.223:3306)
- Web-palvelin: (185.20.137.199)

## **2. YLEISKUVA JÄRJESTELMÄSTÄ**

### **2.1 KUVAUS**

Usein etenkin IT-alalla henkilöstökulut ovat yrityksille suhteellisen suuri ja tärkeä kuluerä – tämän takia on tärkeää tietää työntekijöiden työtunneista ja palkasta mahdollisimman paljon. Ilman IT-järjestelmää tämä voi olla hidasta ja hankalaa. Tämän projektin ydin onkin työntekijöiden tuntikirjanpito, jossa otetaan huomioon erilainen työ (yötyö/pyhäpäivät/normaalit tunnit/asiakkaalle laskutettavat tunnit/jne.) ja näiden perusteella muodostettavat palkat.

Perusidea on, että työntekijät kirjaavat tuntinsa järjestelmään, josta heidän esimiehensä voivat helposti seurata tehtyjä tunteja. Tämän lisäksi ohjelmaan voidaan toteuttaa erilaiset palkkaluokat, työntekijöiden tietojen hallinta ja erilaisia HR-työkaluja – esim. tiimien ja niiden tietojen hallintaan liittyvää toiminnallisuutta. Tarkoituksena on pitää ohjelma helposti laajennettavana ja tehdä sitä iteroiden koko projektin keston ajan.

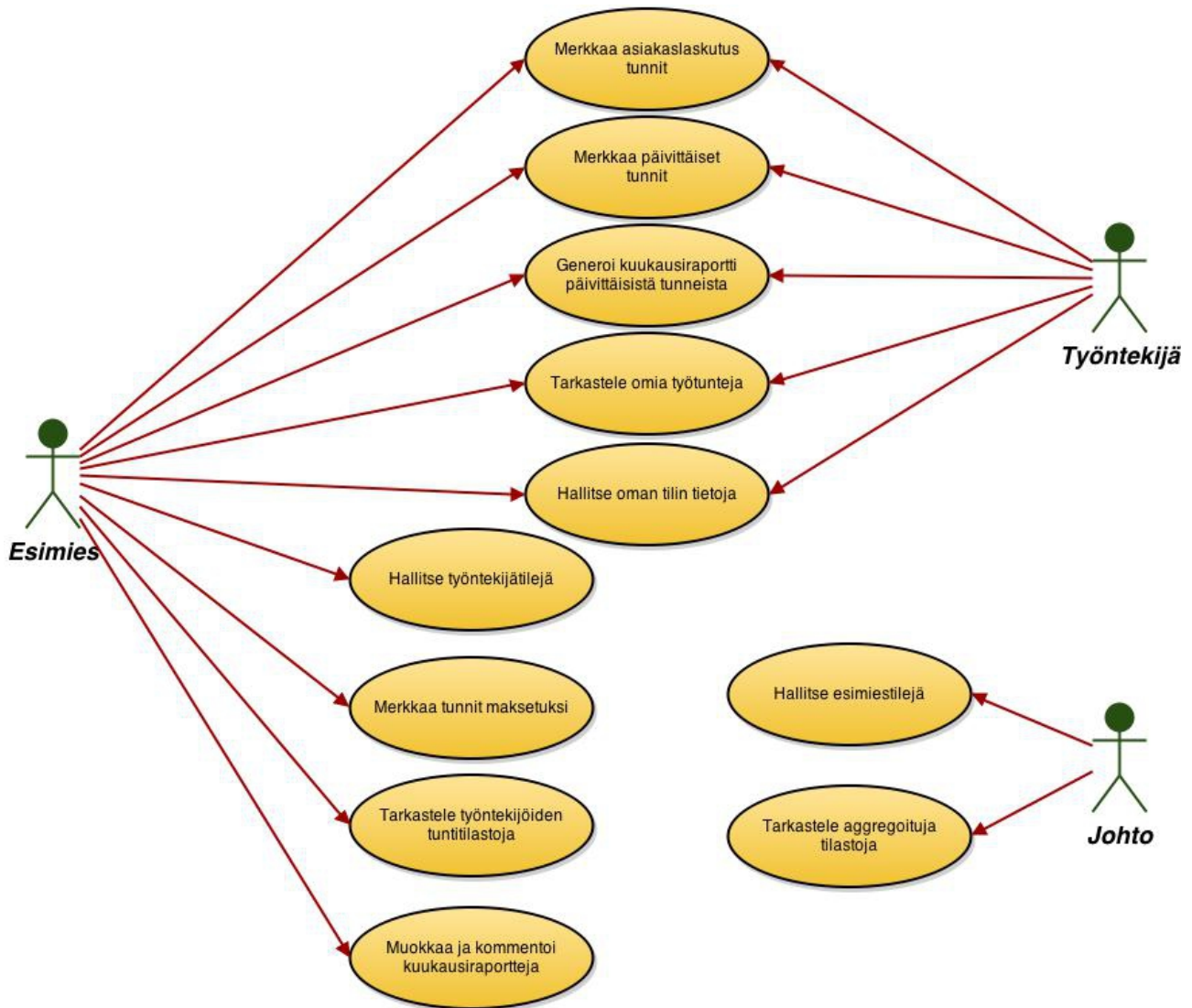
Ohjelmaan voidaan myös merkitä asiakkaalta laskutettavat tunnit, joissa on monesta moneen suhde käyttäjien ja asiakastuntien välillä – asiakaille tehdyissä töissä kuitenkin saattaa olla useampia työntekijöitä.

Ainakin alkuun ohjelma tulee perustumaan hyvin pitkälti CRUD toimintoihin ja resurssien näkymiin (resurssikohtaiset index näkymät). Loppue kohden saatan toteuttaa jotain hienompaa analytiikkaa, mikäli aikaa riittää; ehkäpä toteutan jonkin “number crunching” osuuden palkka- ja työtilastoista ?

## 2.2. KÄYTTÖTAPAUKSET

### 2.2.1 KÄYTTÖTAPAAUSKAAVIO

Käyttötapauskaaviosta on jätetty pois kirjautuminen ja rekisteröityminen, sillä järjestelmään ei kirjautumatta pääse ollenkaan. Rekisteröitymistäkään ei tule, vaan esimiehet voivat luoda käyttäjätilejä työntekijöille (= hallitsevat työntekijätilejä).



### **2.2.2 KÄYTTÄJÄT**

#### **Työntekijä:**

Saa muokata omia tietojaan sekä merkata tehtyjä työtuntejaan.

#### **Esimies:**

Saa muokata työntekijöiden tietoja ja heidän työtuntien tietoja – työntekijälle tulee kuitenkin jättää kopiot kaikista markatuista tunneista, jotta esimies ei voisi väärinkäyttää ohjelmistoa.

#### **Johtoryhmän jäsen:**

Tarkastelee laajempaa analytiikkaa, johon työntekijät eivät pääse käsiksi. Hallitsee esimiestilejä.

## 2.2.3 KÄYTTÖTAPAUSKUVAUKSET

### **Työntekijä ja Esimies yhteiset käyttötapaukset (myös esimies on työntekijä):**

#### Merkkaa asiakaslaskutus tunnit:

Merkataan HTML lomakkeella tunnit, jotka ovat asiakkaalle laskutettua työtä. Monesta moneen suhde käyttäjät – asiakastunnit, koska samaa asiakkaalle laskutettavaa projektia on saattanut olla tekemässä useampi työntekijä.

#### Merkkaa päivittäiset tunnit:

Tunnit merkitään HTML lomakkeella, lomakkeessa on erittelyjä eri kategorioille; yötyö, asiakkaalle laskutettava työ, normaalit työtunnit, jne.

#### Generoi kuukausiraportti päivittäisistä tunneista:

Palkka maksetaan aina kuukauden tuntien perusteella, eli käyttäjän pitää hyväksyä kuukauden työtunnit yhtenä kokonaisuutena; ohjelma osaa generoida kuukausiraportin hyväksymistä varten päivittäisistä tunneista.

#### Tarkastele omia työtunteja:

Työntekijä voi tarkastella omia merkattuja tuntejaan.

#### Oman tilin tietojen hallinta:

Omassa käyttäjätilissä on työntekijän palkanmaksuun tarvittavat tiedot, jotka voivat luonnollisesti muuttua esim. muuton tai pankin vaihtamisen takia.

### **Esimiehen käyttötapaukset (pl. edellä mainitut):**

#### Hallitse työntekijätilejä:

CRUD operaatiot työntekijätileihin.

#### Merkkaa tunnit maksetuksi:

Merkkaa tunnit maksetuksi, jolloin työntekijät tietävät, että tunnit ovat käsitelty ja maksettu.

#### Tarkastele työntekijöiden tunteja:

Tarkastele yksittäisten työntekijöiden tuntimerkintöjä ja kuukausiraportteja. Työntekijähän näkevät vain omat tuntinsa, mutta esimies näkee kaikkien työntekijöiden.

#### Muokkaa/kommentoi kuukausiraportteja:

Esimies voi muokata ja kommentoida kuukausiraportteja jos niissä on jotain epäselvyyksiä; työntekijöiden alkuperäiset merkinnät pitää kuitenkin myös selvittää tallella, ettei esimies voi vahingossa tai tahallaan muuttaa näitä työntekijän edun vastaisesti.

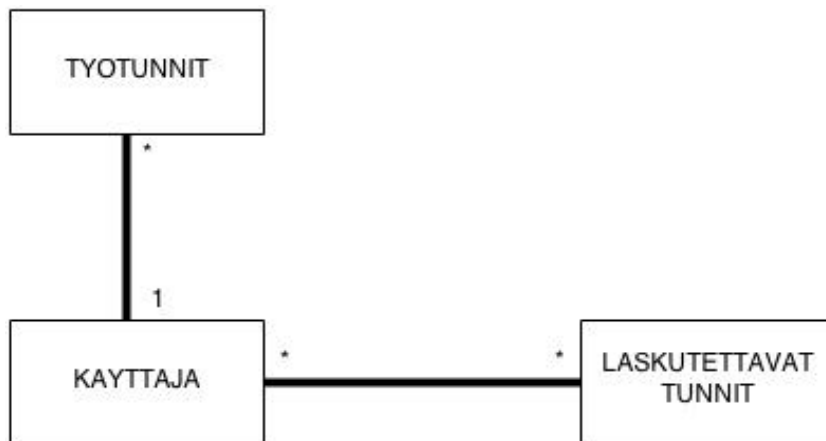
### **Johto:**

#### Hallitse esimiestilejä:

#### Tarkastelee aggregoituja tilastoja:

Kuukausiraportteista voidaan generoida aggregoituja raportteja työvoimakuluista, joita voidaan käyttää apuna esim. työn tuottavuuden mittaamisessa ja rekrytointitarpeiden selvittämisessä.

### 3. TIETOSISÄLTÖ



Tietokohde: Käyttäjä

Attribuutti	Arvojoukko	Kuvailu
Id	Int	Uniikki käyttäjäid
Username	String, 16	Käyttäjänimi
Password	String, hashattu	Käyttäjän salasana
Firstname	String, 32	Etunimi
Lastname	String, 32	Sukunimi
Address	String, 128	Osoite, vapaamuotoinen... käyttäjän omalla vastuulla tulevatko palkkakuitit perille.
Email	String, 254	Sähköposti, spostin max pituus 254
Phone	String, 15	Puhelin
Rank	Int, 0,1,2	"status", esimies, työntekijä, johtaja...

Tietokohde: Työtunnit

Attribuutti	Arvojoukko	Kuvailu
Id	Int	Uniikki id
Day	Date	Päivämäärä, minkä päivän tunnit
Hours	Int	Normaalit tunnit 6-22
Offhours	Int	Muut 22-6
Standbyhours	Int	Päivystystunnit
Deleted	Int	Onko poistettu?
User_id	Int	Viittaus käyttäjään, jonka tunneista kyse

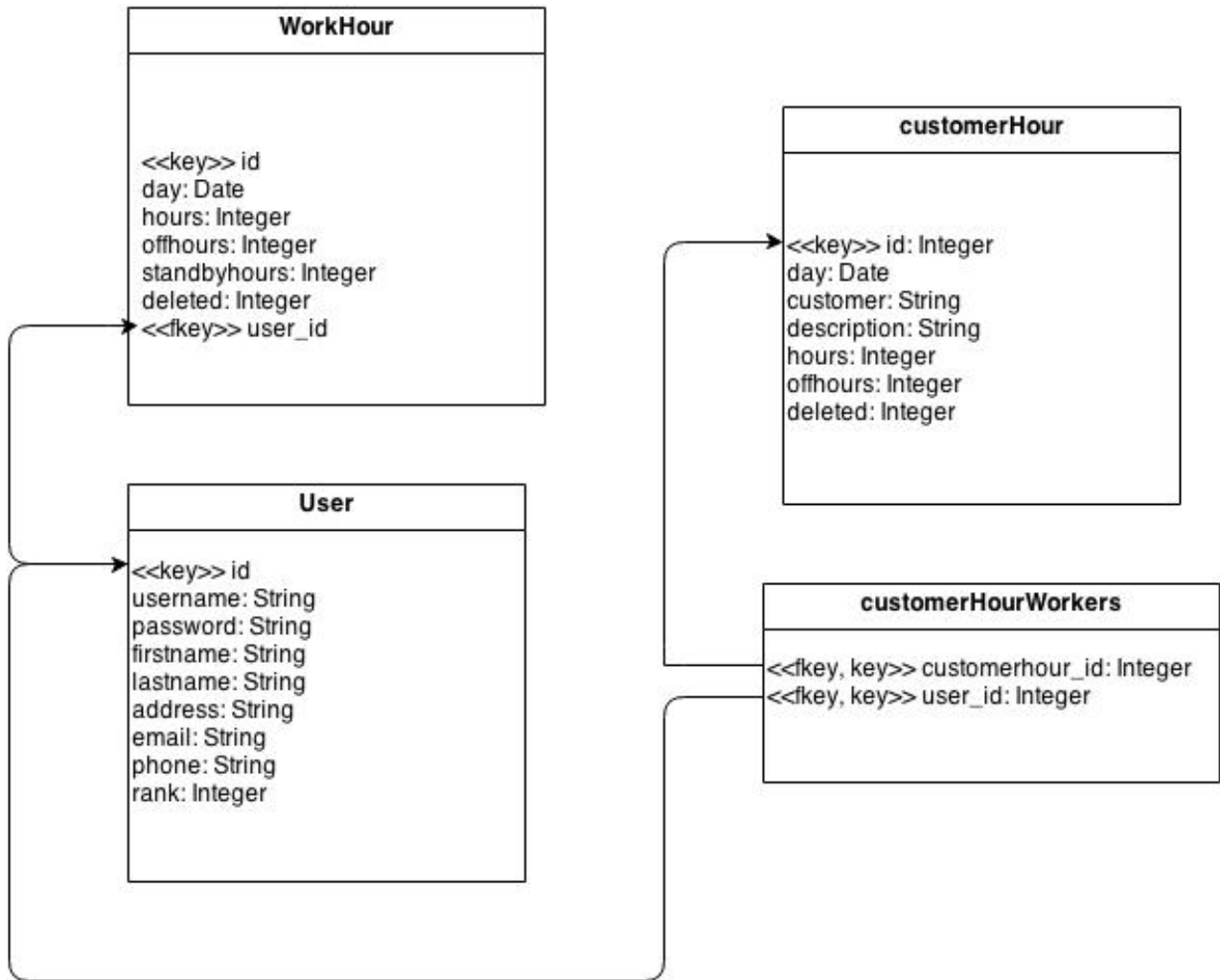
Tietokohde: Laskutettavat tunnit (asiakastunnit)

Attribuutti	Arvojoukko	Kuvailu
Id	Int	Uniikki id
Day	Päivämäärä Date	Minä päivänä tehtyjä tunteja?
Customer	String, max 20	Asiakkaan nimi
Description	Text	Kuvaus, lisätiedot. Vapaamuotoinen.
Hours	Int 0-16	Tunnit 6-22
Offhours	Int 0-8	Tunnit 22-6
Deleted	0,1	Onkopoistettu?

Monesta moneen yhteys asiakastuntien ja käyttäjien välillä toteutetaan välitaululla, jossa on vain viittaus kumpaankin - ks. Relaatiokaavio.



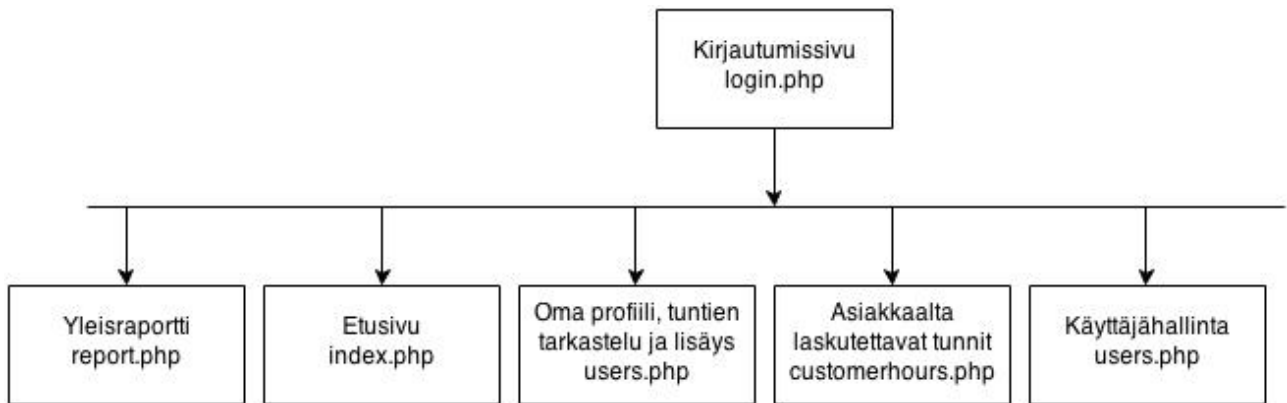
#### 4. RELAATIOKAAVIO



## 5. JÄRJESTELMÄN YLEISRAKENNE

Järjestelmä noudattaa MVC mallia, jossa tietokantayhteys ja joitain apufunktioita on asetettu /lib kansioon. Järjestelmän kaikki pyynnot ohjataan index.php tiedostoon, joka hoitaa autentikoinnin ja pyyntöjen reitityksen kontrollereille.

## 6. KÄYTTÖLIITTYMÄ



Järjestelmään ei pääse ollenkaan sisälle ilman kirjautumista, tämän jälkeen eri osioihin pääsee navigointipalkista yllä esitetyllä tavalla. Pyrin lisäämään CRUD toiminallisuuden suoraan tuntien/asiakastuntien/käyttäjien jne. "index" näkymiin, ettei erillisiä muokkaus ja lisäyssivuja tarvitsisi olla – tämän on tarkoitus helpottaa järjestelmän käyttöä, sillä tarkoituksena on olla mahdollisimman helppokäyttöinen ja kevyt sovellus.

## 7. ASENNUSTIEDOT

Oletetaan, että asentajalla on LAMP stack toiminnassa. Tällöin riittää vetää sovelluksen GitHub repo ja laittaa CONFIG.php tiedostoon tietokannan tunnukset. Tietokannan saa pystyyn SQL lauseilla, jotka löytyvät kansioista /sql/.

Huom. .htaccess on erittäin tärkeä sovelluksen toimivuudelle etenkin siltä osalta, että se ohjaa kaikki pyynnöt index.php tiedostolle reititettäväksi.

Tällä hetkellä sovelluksen tietokanta on eri palvelimella kuin itse sovellus ja tietokantayhteys tapahtuu julkiverkon yli (voin käyttää samaa kantaa useassa devausympäristössä) – oikeassa tilanteessa tietokantapalvelimeen yhdistettäisiin tietenkin lähiverkon yli (kun samassa pilvessä ovat) tai localhost kautta, jos kanta on samalla koneella.

## 8. KÄYTTÖOHJEET

Sovellusta on mahdoton käyttää ilman sisäänkirjausta, sillä .htaaccess ohjaa kaiken liikenteen index.php tiedostoon, joka taas ei päästä etenemään ellei ole kirjautunut sisään.

Sisäänkirjautumisen (testitunnus u: user1, p: user1) jälkeen sovelluksen käyttö on melko vaivatonta; listausnäkymistä voi suoraan tehdä CRUD operaatiot Bootstrap3 modalin kautta ('edit' avaa tämän).

HUOM! Tällä hetkellä CRUD operaatiot toteuttu vain "Asiakastunnit" osalta. Näissä myös monesta moneen suhde käyttäjiin, ja syötteeksi pitääkin antaa valideja käyttäjiä pilkulla eroteltuna.