

Tietokantasovellus Vuorolistajärjestelmä

Tuukka Ylönen
Johanna Keto

Kesä 2013

Johdanto

Toteutettava ohjelmamme on nimeltään Vuorolistajärjestelmä, jota käytetään kotihoidon yksikössä. Kotihoidossa on x määrä hoitajia ja y määrä asiakkaita, joilla on 0-5 käyntiä päivässä. Kullakin käynnillä on tietty kesto, joka vaihtelee asiakkaittain. Vuorolistajärjestelmän tarkoituksena on jakaa yksittäiset asiakaskäynnit vuorossa olevien hoitajien kesken niin, että työtaakka on kaikkien kesken mahdollisimman tasainen. Vuorot jaetaan saman tien koko viikoksi.

Tietokantaan talletetaan asiakkaiden tiedot sekä käynnit ja niiden kestot. Tietokannasta löytyy myös hoitajat ja kunkin hoitajan lupatiedot eli tieto esimerkiksi siitä, voiko hoitaja jakaa lääkkeitä. Myös käynteihin talletetaan lupatiedot, jotta sopiva hoitaja voidaan yhdistää sopivaan käyntiin.

Toteutus/toimintaympäristö

Työ on toteutettu python3-ohjelmointikielellä käyttäen Bottle-webkehystä. Ohjelmisto vaatii python3-kieltä sekä virtualenv-ohjelmaa, jonka avulla asennetaan muut tarvittavat kirjastot. Selaimelta ei vaadita perus html- ja css tukea kummempia. Tietokantana käytetään sqlite3:a, mutta se asentuu asennusohjeissa luotavaan virtualenv-ympäristöön.

Käyttötapaukset

Sisäänkirjautuminen:

Rekisteröitynyt käyttäjä voi kirjautua sisään ohjelmaan. Ennen sisäänkirjautumista ohjelmassa ei voi tehdä mitään muuta kuin rekisteröityä.

Rekisteröityminen:

Kuka tahansa voi rekisteröityä. Rekisteröityessä tulee valita tunnus, joka ei ole vielä käytössä ja salasana tulee kirjoittaa kahdesti.

Uuden hoitajan lisääminen:

Rekisteröitynyt käyttäjä voi lisätä ohjelman tietokantaan uuden hoitajan, jolle vuoroja voidaan jakaa. Hoitajalle määritellään nimi ja luvat.

Uuden asiakkaan lisääminen:

Rekisteröitynyt käyttäjä voi lisätä ohjelman tietokantaan uuden asiakkaan, jolle määritellään nimi.

Uuden käynnin lisääminen:

Rekisteröitynyt käyttäjä voi lisätä valitsemalleen asiakkaalle uuden käynnin. Käynnin yhteydessä valitaan, mitä lupia sillä käyvällä hoitajalla tulee olla voimassa.

Asiakkaan poistaminen tietokannasta:

Rekisteröitynyt käyttäjä voi poistaa asiakkaan tietokannasta, kun tälle ei tehdä enää käyntejä.

Hoitajan poistaminen tietokannasta:

Rekisteröitynyt käyttäjä voi poistaa hoitajan tietokannasta, jos tämä ei enää jatkossa tee käyntejä.

Käynnin poistaminen tietokannasta:

Rekisteröitynyt käyttäjä voi poistaa turhan käynnin asiakkaan tiedoista painamalla käynnin perässä olevaa ”poista”-nappia.

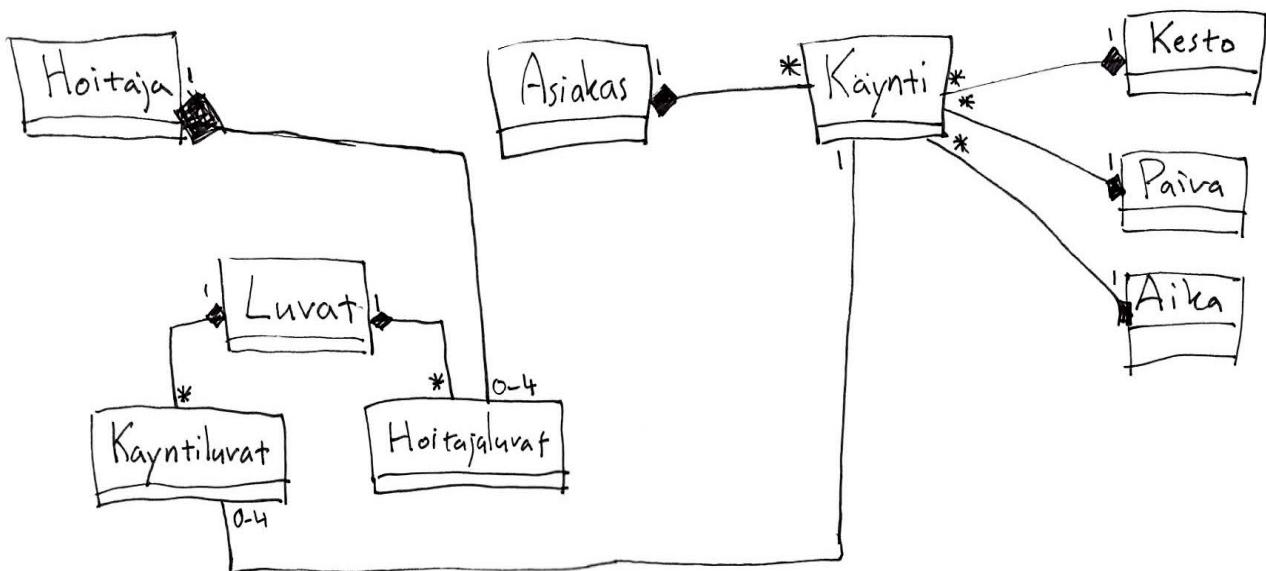
Tietosisältö

Tietokantaan kuuluvat taulut Käyttäjä, Hoitajat, Asiakkaat, Käynnit, Luvat.

Lupiin liittyvät taulut Hoitajaluvat ja Asiakasluvut.

Lisäksi Käyntiin liittyvät taulut Kestot, Ajat ja Päivät.

tietosisältökaavio



Tietokohteiden määrittely:

Asiakas

Attribuutti	Arvojoukko	Kuvaus
ID	integer	pääavain
nimi	merkkijono	Asiakkaan nimi

Asiakkaalla voi olla monia käynnejä. Käynneille määritellään lupavaatimukset erikseen.

Hoitaja

Attribuutti	Arvojoukko	Kuvaus
ID	integer	pääavain
nimi	merkkijono	Hoitajan nimi

Hoitajalla voi olla 0-4 lupaa.

Käynti

Attribuutti	Arvojoukko	Kuvaus
ID	integer	pääavain
asiakasID	integer	Yhdistää Asiakkaaseen
paivaID	integer	Yhdistää Päivään
aikaID	integer	Yhdistää Aikaan
kestoID	Integer	Yhdistää Kestoon

Käynnillä voi olla 0-4 lupaa. Luvat määrittelevät, mitä lupia hoitajalla tulee olla voidakseen käydä käynnillä. Käynti liittyy yhteen asiakkaaseen, päivään, aikaan ja keston. Päivä- Aika ja Kestotaulut yhdistävät iideet selkokielesiin arvoihin ja varmistavat tietokannan eheyttä.

Luvat

Attribuutti	Arvojoukko	Kuvaus
ID	integer	pääavain
lupa	merkkijono	Haavat, lääkkeet(lääkejako), piikit ja silmätipat

Lupia on neljää eri sorttia: haavanhoitolupa, lääkkeidenjakolupa, piikinantolupa sekä silmätippojen laittolupa. Yksi Käyntilupa tai Hoitajalupa yhdistyy aina yhteen Lupaan.

Käyntiluvat

Attribuutti	Arvojoukko	Kuvaus
ID	integer	pääavain
lupaID	integer	Yhdistää Lupaan

Käyntilupa kertoo, mitä lupia käyntiin liittyy.

Hoitajaluvat

Attribuutti	Arvojoukko	Kuvaus
ID	integer	pääavain
lupaID	integer	Yhdistää Lupaan

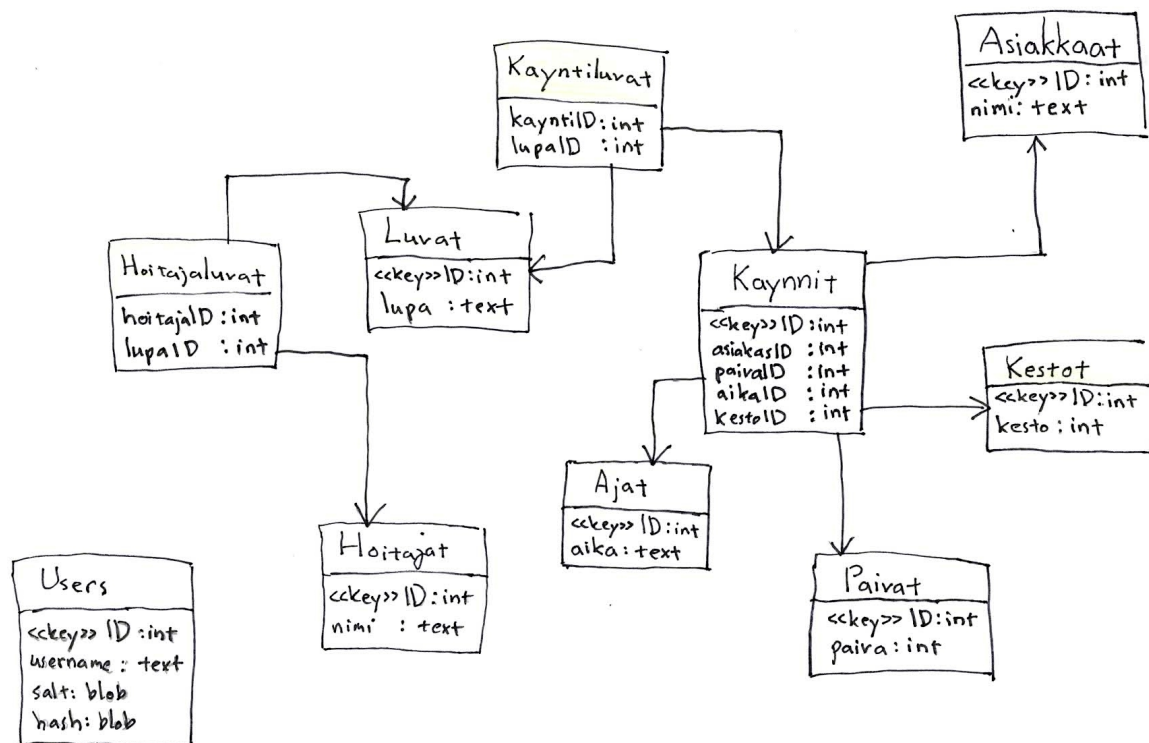
Hoitajalla voi olla 0-4 lupaa. Lupa kertoo, minkälaisilla käynneillä hoitaja voi käydä.

Users

Attribuutti	Arvojoukko	Kuvaus
ID	integer	pääavain
username	Merkki	Käyttäjänimi
Salt	Blob	Suola salasanalle
Hash	blob	Tiiviste suolasta ja salasanasta

Käyttäjänimi, luodaan rekisteröitymisessä. Yksilöllinen.

Relaatiotietokantakaavio



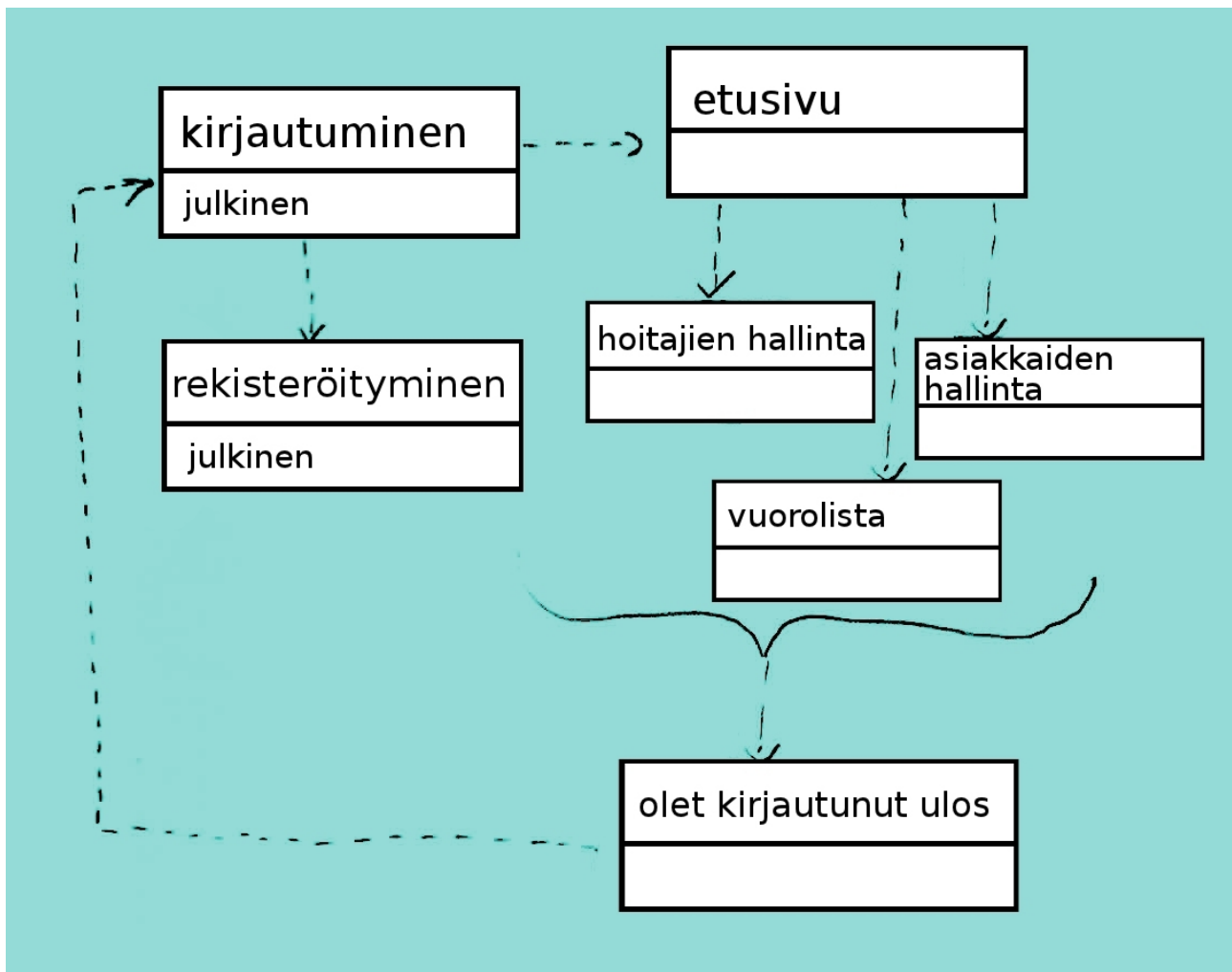
Käyttöliittymä

Aloitussivuna toimii sisäänkirjautuminen, joka ohjaa tunnuksettoman käyttäjän rekisteröitymis-sivulle. Rekisteröitymisen jälkeen käyttäjä voi kirjautua ja pääsee sitten minne vain ohjelmassa. Etusivulta pääsee Hoitajien hallintaan, jossa voi lisätä ja poistaa uusia hoitajia, sekä Asiakkaiden hallintaan, jossa voi lisätä ja poistaa asiakkaita ja uusia hoitokäyntejä.

Etusivulta pääsee myös ohjelman pääominaisuuteen, eli vuorolistasivulle, johon listataan kunkin hoitajan viikon hoitokäynnit.

Kaikkien sivujen yläpalkissa on linkit ”logout” ja ”etusivulle”. Logout-sivulta pääsee takaisin etusivulle, joka ohjaa kirjautumiseen.

Käyttöliittymäkaavio



Järjestelmän yleisrakenne

Järjestelmä noudattaa löyhästi MVC-mallia, näkymät löytyvät views-hakemistosta ja kontrollerit löytyvät routes.py-tiedostosta. Mallia ohjelmassa on tietosisällön ja siihen liittyvien säiliöluokkien (asiakas.py, kaynti.py, hoitaja.py, user.py) lisäksi kuitenkin sen verran vähän että hoitovuorojen jako tapahtuu routes.py-tiedostossa.

Static-hakemistosta löytyy staattiset tiedostot ja Dokumentaatio-hakemistossa on dokumentaatioon liittyviä tiedostoja.

Muut hakemistot ovat virtualenvin luomia ja sisältävät kirjastoja ja python-ympäristöön liittyviä tiedostoja.

Järjestelmän komponentit

Selaimessa selattavat pydoc-dokumentaatiot saa komennolla pydoc -p 1234.

Pydocista löytyy kutakuinkin samat tiedot kuin tästäkin.

Komponentit ovat tässä listattuna suunnilleen siinä järjestyksessä, missä niihin kannattaa tutustua.

main.py

Ohjelman aloituskohta. main.py ottaa komentoriviltä asetukset, luo tietokantaoliot ja Bottle-ohjelman jonka se myös laittaa käyntiin.

routes.py

routes.py sisältää Bottlen handlerit eri osoitteille

create_tables.py

Skripta tyhjän tietokannan luomiseen, tätä voi käyttää stand-alonena tai kutsua muusta koodista (main.py kutsuu tätä jos tietokantaa ei ole)

sqlite_backend.py

Tämä moduuli sisältää llioita sql-tietokannan käyttöä varten.

auth.py

Sisältää Auth-luokan jolla hallitaan käyttäjiä ja niiden autentikointia.

hoitaja.py, asiakas.py, kaynti.py, user.py

sisältävät säiliöoliot hoitajille, asiakkaille, käynneille ja käyttäjille.

views-hakemiston komponentit:

views-hakemisto sisältää Bottlen SimpleTemplate Enginen muodossa olevia html-template-tiedostoja. Ohjelman yksittäiset sivut koostuvat useasta templatesta joten templateja on hyvin paljon. Tässä tärkeimmät:

layout.tpl, loginlayout.tpl:

Sivujen layoutin määrittävät templatet. layout.tpl-templatea käytetään kaikkiin sisäänkirjautumista vaativiin sivuihin ja loginlayouttia käytetään login-sivuun sekä käyttäjän rekisteröintisivuun.

hoitajat.tpl:

Hoitajien listaus ja lisäys -sivu. Käyttää avukseen poistaHoitaja.tpl-templatea hoitajien poistamiseen.

asiakkaanHallinta.tpl:

Asiakkaiden ja niihin liittyvien hoitovuorojen listaus, lisäys ja poistosivu. Käyttää apunaan templateja asiakkaanLisays.tpl, lisääVuoro.tpl sekä asiakasKayntiListaus.tpl.

hoitovuorot.tpl:

Näyttää hoitajille jaetut hoitovuorot.

Sovelluksen käyttöönotto:

Tarvitset seuraavat sovellukset:

virtualenv
python 3

Näin käynnistät Vuorolistajärjestelmä-ohjelman:

kirjoita terminaaliin seuraavat
git clone git://github.com/joketo/Vuorolistajarjestelma.git
cd Vuorolistajarjestelma
virtualenv --no-site-packages -p python3 ./
source bin/activate
pip install -r requirements.txt

1. Täysin uusi tietokanta:

python main.py

tai

2. Esimerkkietokanta

Esimerkkietokantaan on talletettu valmiiksi käyttäjä Pekka, kourallinen hoitajia ja asiakkaita, sekä käyntejä asiakkaille. Se sopii mainiosti testailupohjaksi.

Esimerkkietokannan saat käyttöösi kirjoittamalla komentoriville

python main.py -db esimerkki.db

Nyt voit avata sovelluksen selaimellasi <http://localhost:8080/>

Kirjaudu sisään tunnuksella Pekka (salasana: salasana) tai rekisteröidy uudella tunnuksella.