

Invoicing-laskutusjärjestelmän dokumentaatio

Invoicing on www-pohjainen laskutusjärjestelmä, jolla on helppoa luoda uusia laskuja ja tulostaa niitä asiakkaille.

Järjestelmä on pienyrityksen käyttöä varten luotu ja siten sen ominaisuudet on varsin pienet, mutta riittävät, sillä laskutuksessa olisi tarkoitus käyttää mahdollisimman vähän aikaa.

Järjestelmävaatimukset

Palvelimen järjestelmävaatimukset

- WWW-palvelin (esim. Nginx tai Apache)
- PHP 5.6
- PostgreSQL 9.4

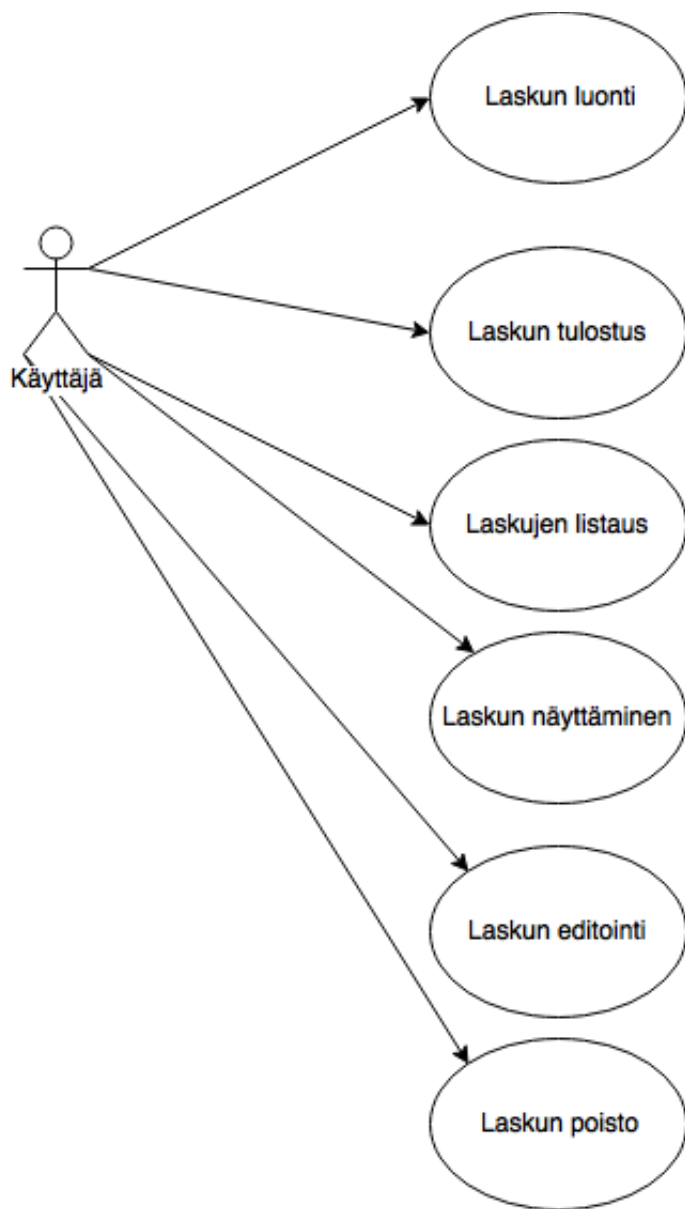
Käyttäjän järjestelmävaatimukset

Käyttäjän selain kuuluu olla nykyaikainen ns. HTML5-tekniikoita tukeva selain.

Käyttäjärühmät

Käyttäjä Käyttäjä on henkilö jolla on täysi käyttöoikeus invoicing-laskutuspalvelussa.

Käyttötapaukset



Laskun luonti

Laskun luonnissa käyttäjä kirjaa uuden laskun lisäämällä asiakastiedot, tuotteet ja niiden hinnat. Järjestelmä laskee laskun loppusumman, generoi laskulle laskunumeron ja viitenumeron.

Laskun tulostus

Käyttäjä voi tulostaa luomansa laskun.

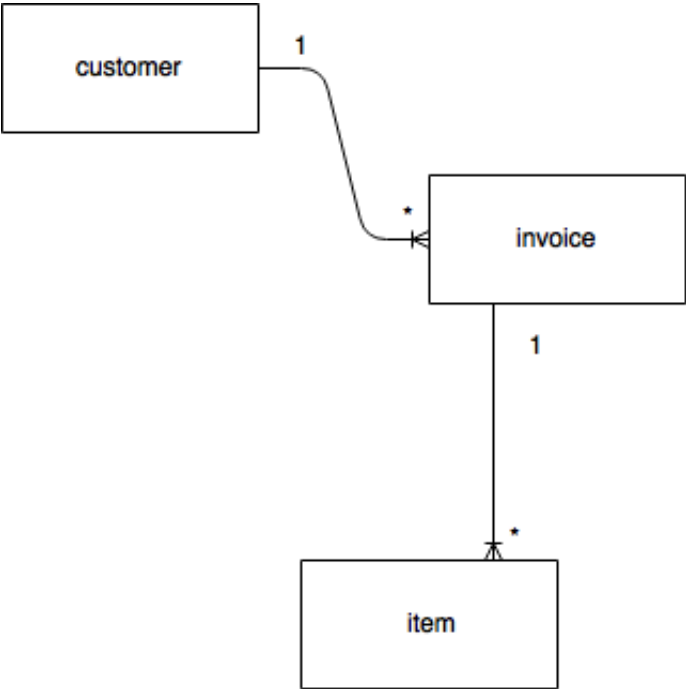
Laskun muokkaus

Käyttäjä voi muokata laskua, lisätä, poistaa tai muokata tuotteita, muokata asiakastietoja, eräpäiviä tms.

Laskun poisto

Käyttäjä voi poistaa laskun.

Järjestelmän tietosisältö



Customer

Attribuutti	Arvojoukko	Kuvailu
id	Kokonaisluku	Primääriavain
name	Merkkijono max 255 merkkiä	Asiakkaan nimi
street_name	Merkkijono max 255 merkkiä	Asiakkaan katuosoite
post_code	Merkkijono max 10 merkkiä	Asiakkaan postinumero
city	Merkkijono max 255 merkkiä	Asiakkaan kaupunki
email	Merkkijono max 255 merkkiä	Asiakkaan sähköposti

Customer on asiakas jotka ovat laskujen maksajia.

Item

Attribuutti	Arvojoukko	Kuvailu
id	Kokonaisluku	Primääriavain
invoice	Kokonaisluku	Foreign key, lasku jolle rivi kuuluu
description	Merkkijono max 255 merkkiä	Tuotteen kuvaus
amount	Kokonaisluku	Tuotteiden määrä
price	Kokonaisluku	Tuotteen hinta ALV 0% /kpl, sentteinä
tax	Kokonaisluku	Maksettava ALV

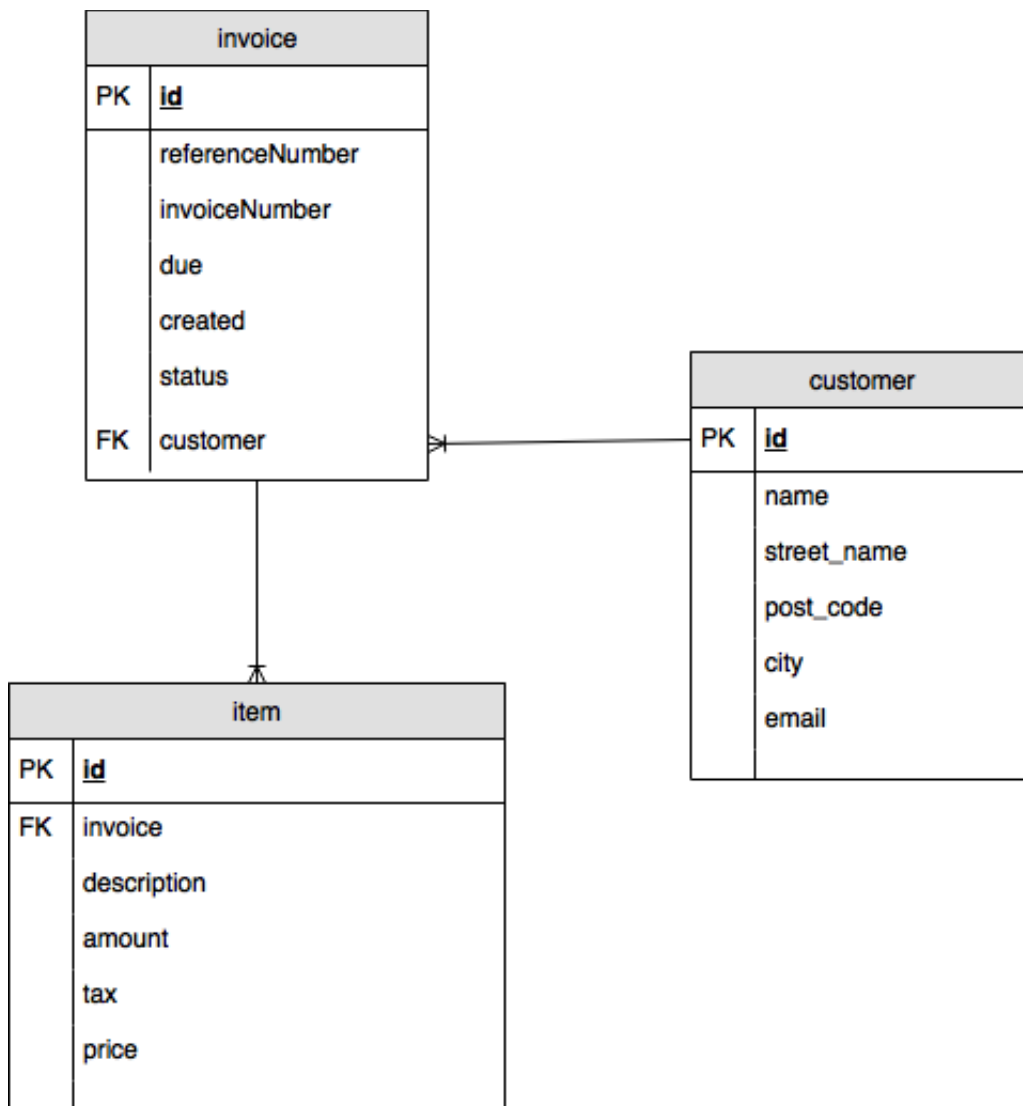
Item on laskurivi.

Invoice

Attribuutti	Arvojoukko	Kuvailu
id	Kokonaisluku	Primääriavain
invoice_number	Kokonaisluku	Laskun numero
reference_number	Kokonaisluku	Laskun viitenumero
created	Päivämäärä	Päivä jolloin lasku luotu
due	Päivämäärä	Laskun eräpäivä
status	Enum	Laskun tila
customer	Kokonaisluku	Foreign key, viittaus asiakkaaseen

Invoice on lasku, joka luodaan asiakkaalle. Status tarkoittaa laskun tilaa, vaihtoehtoina on PENDING tai PAID.

Relaatiotietokantakaavio



Järjestelmän yleisrakenne

Järjestelmä on toteutettu Angular 2 -frontend javascript-kirjastolla ja Symfony 3.0 -backendillä.

Angular-ja Html-lähdekoodit löytyvät frontend -hakemistosta. Frontend-koodi kirjoitetaan ylläolevaan hakemistoon ja ajetaan Gulp task-runnerin läpi web -hakemistoon.

Symfony:n osalta järjestelmä t”ottelee ns. Standard edition hakemistorakennetta. src -hakemistossa on backend-lähdekoodi. Tämä lähdekoodi on kirjoitettu PSR-4 standardeja mukaillen.

Ohjelman controllerit ovat Invoicing\Bundle\AppBundle\Controller -hakemistossa. Sovelluksen modelit ovat Invoicing\Model -hakemistossa. Sovelluksen servicet ovat Invoicing\Service -hakemistossa.

Järjestelmän asennus

Järjestelmä vaatii seuraavat komponentit toimiakseen:

- npm
- php 5.6
- postgresql 9.4
- yleisimmät php-moduulit esim. php-pdo, php-postgresql, php-intl.

Ohjelmiston asennus:

1. Ladataan ohjelmistokoodi Githubista
2. Luo ohjelmistolle tietokanta
3. Ajetaan `composer install`. Täytä kysymyksiin vastaukset, tämä luo automaattisesti konfiguraatiodoston ympäristöllesi.
4. Aja tietokantamigraatit komennolla: `php bin/console phinx:migrations:migrate`
5. Osoita WWW-palvelin (esim nginx) osoittamaan 'web/' -hakemistoon (katso konfiguraatiodosto `doc/nginx.example.conf`)
6. Ohjelmisto on valmis käytettäväksi

Vaihtoehtoinen tapa luoda tietokanta on importtaamalla `create_tables.sql`. Tällöin ei tosin ole migraatioita käytössä.

Testaus

Ohjelmistoa on testattu käsipelillä. Valmius Unit ja Integraatitestaukseen on mutta ohjelmiston yksinkertaisuuden, ORM:in puuttuminen aiheutti sen etten lähtenyt tekemään automatisoituja testejä. Integraatitestejä varten olisi tarvinnut kunnollisen tavan tehdä testidataa, mikä ei ollut järin yksinertaista ilman vaadittuja kirjastoja.