

Pizza-Palvelu

Johdanto:

Tarkoituksena on tehdä pizzerian varausjärjestelmä ja asiakkaiden hallintasovellus. Sovellus toimii pizzerian tarkoitusten mukaisesti varaus, laskutus, asiakkaiden tietokantana. Sovellus mahdollistaa pizzojen tilaamisen verkko-sivuston kautta. Sovellus mahdollistaa tietoverkkojen käytön pizzan tilaamisessa.

Totetus ja toiminta-ympäristö ovat Linux -pohjaisten serveri-ratkaisun käyttö, ja Apache -palvelin sekä PostgreSQL -tietokanta-ohjelmiston varaan rakennettu sovellus. Sovellus toimii *armv7* -pohjaisessa palvelimessa. Sovellus kehitetään käyttäen normaaleja UNIX -pohjaisia sovelluksia kuten *vi* -editori.

Sovellus toteutetaan PHP -ohjelmointikielessä. Sovellus käyttää **Twig** -nimistä template -engineä. Sovellus käyttää **SLIM** -nimistä sovelluskehystä PHP:n päälle. Myös **PDO** -kirjastoja käytetään tiedon hakemista ja varastointia varten tietokantaan. Sovellus on suunniteltu alusta asti tietoturvalliseksi, esim. käyttäjien syötteet normalisoidaan aina. Sovellus käyttää Cookie:ita tukeakseen sessioita. Sovellus tehdään mahdollisimman vähän Javascript -kieltä käyttäen. Myöskään mitään Flash, HTML5, tai vastaavia sovelluksia ei käytetä. Sovellus toimii yhtä tietokantaa käyttäen.

Käyttötapaukset:

1. Järjestelmään rekisteröityminen

Käyttäjät: asiakas

Tavoite: saada käyttäjätunnus järjestelmään

Käyttötapauksen kulku:

Käyttäjä valitsee käyttäjätunnuksen ja sähköposti-osoitteen. Järjestelmä lähettää salasanan sähköpostitse käyttäjälle.

2. Tuotteiden selaaminen

Käyttäjät: asiakas

Tavoite: valita haluamansa tuote

Käyttötapauksen kulku:

Käyttäjä kirjautuu järjestelmään. Käyttäjälle esitetään listat tuotteista.

3. Tilauksen laatiminen

Käyttäjät: asiakas

Tavoite: tilata pizza

Käyttötapauksen kulku:

Käyttäjä valitsee haluamansa tuotteet. Sovellus esittää yhteenvedon tilauksesta.

Käyttäjä vahvistaa tilauksensa.

Poikkeuksellinen toimita:

Käyttäjällä häiriöitä, jonka tapauksessa tilaus voidaan peruttaa.

4. Kuitin tulostus

Käyttäjät: asiakas

Tavoite: saada kuitti tilauksesta

Käyttötapauksen kulku:

Käyttäjä valitsee kuitin -tulostuksen joka esitetään PDF -tiedostona käyttäjälle.

5. Kuitin tulostus

Käyttäjät: ylläpito

Tavoite: kuitti tilauksesta

Käyttötapauksen kulku:

Pizza-palvelu tulostaa PDF -dokumentin kuitista toimitettavaksi toimituksen yhtydessä.

6. Ylläpidon järjestelmään kirjautuminen

Käyttäjät: ylläpito

Tavoite: Pizza-palvelun ylläpito

Käyttötapauksen kulku:

Käyttäjä kirjautuu järjestelmään pääkäyttäjänä etukäteen annetulla salasanalla.

7. Asiakkaiden tietojen selaaminen

Käyttäjät: ylläpito

Tavoite: hallinnoida asiakkaiden tietoja

Käyttötapauksen kulku:

Käyttäjä kirjautuu järjestelmään. Käyttäjä voi hallinnoida asiakkaiden tietoja.

8. Tilausten selaaminen

Käyttäjät: ylläpito

Tavoite: hallinnoida tilauksia

Käyttötapauksen kulku:

Käyttäjä kirjautuu järjestelmään. Käyttäjä voi hallinnoida aviomia ja suljettuja tilauksia.

9. Tilauslistan tulostus

Käyttäjät: ylläpito

Tavoite: tulostaa tilauksista tietoja

Käyttötapauksen kulku:

Käyttäjä kirjautuu järjestelmään. Käyttäjälle esitetään suljettujen ja avointen tilausten listat PDF -dokumenttina.

10. Aikakohtaisten hintojen lisäys, poisto ja muokkaus

Käyttäjät: ylläpito

Tavoite: ylläpitää aikaan liittyviä hintoja

Käyttötapauksen kulku:

Käyttäjä kirjautuu järjestelmään. Käyttäjä voi luoda, poistaa tai muokata aikakohtaisia tarjouksia.

11. Toimitusten kirjaus

Käyttäjät: ylläpito

Tavoite: lisätä tieto toimituksesta

Käyttötapauksen kulku:

Käyttäjä voi merkitä tilauksen toimitetuksi. Myös häiriö-merkinnät tehdään tässä.

12. Myöhästymisalennuksen kirjaus

Käyttäjät: ylläpito

Tavoite: merkitä kantaan annettu alennus myöhästyneestä toimituksesta

Käyttötapauksen kulku:

Käyttäjä merkitsee järjestelmään myöhästyneen toimituksen. Asiakas voi halutessaan perua tai muuttaa yli tunnin kestänyttä toimitusta.

13. Tarjousten kirjaus

Käyttäjät: ylläpito

Tavoite: merkitä järjestelmään tarjouksia

Käyttötapauksen kulku:

Käyttäjällä on mahdollisuus kirjata uusia tarjouksia, peruttaa tarjouksia tai muokata olemassaolevia.

14. Tilauksen peruminen häiriöiden vuoksi

Käyttäjät: ylläpito

Tavoite: peruuttaa tilaus jos asiakkalla häiriömerkintöjä

Käyttötapauksen kulku:

Käyttäjän häiriömerkinnät näkyvät käyttäjä-listassa sekä tilausvahvistuksessa, ja käyttäjä voi poistaa/peruuttaa tilauksen.

15. Tuottetietojen lisäys, muokkaaminen ja poisto

Käyttäjät: ylläpito

Tavoite: hallinnoida tuottetietoja

Käyttötapauksen kulku:

Käyttäjä voi muokata olemassaolevia tuottetietoja, poistaa niitä ja lisätä uusia.

16. Lisukkeiden lisäys, muokkaaminen ja poisto

Käyttäjät: ylläpito

Tavoite: hallinnoida lisuketietoja

Käyttötapauksen kulku:

Käyttäjä voi muokata olemassaolevia lisuke-tietoja, poistaa niitä ja lisätä uusia.

17. Tuoteryhmien hallinta

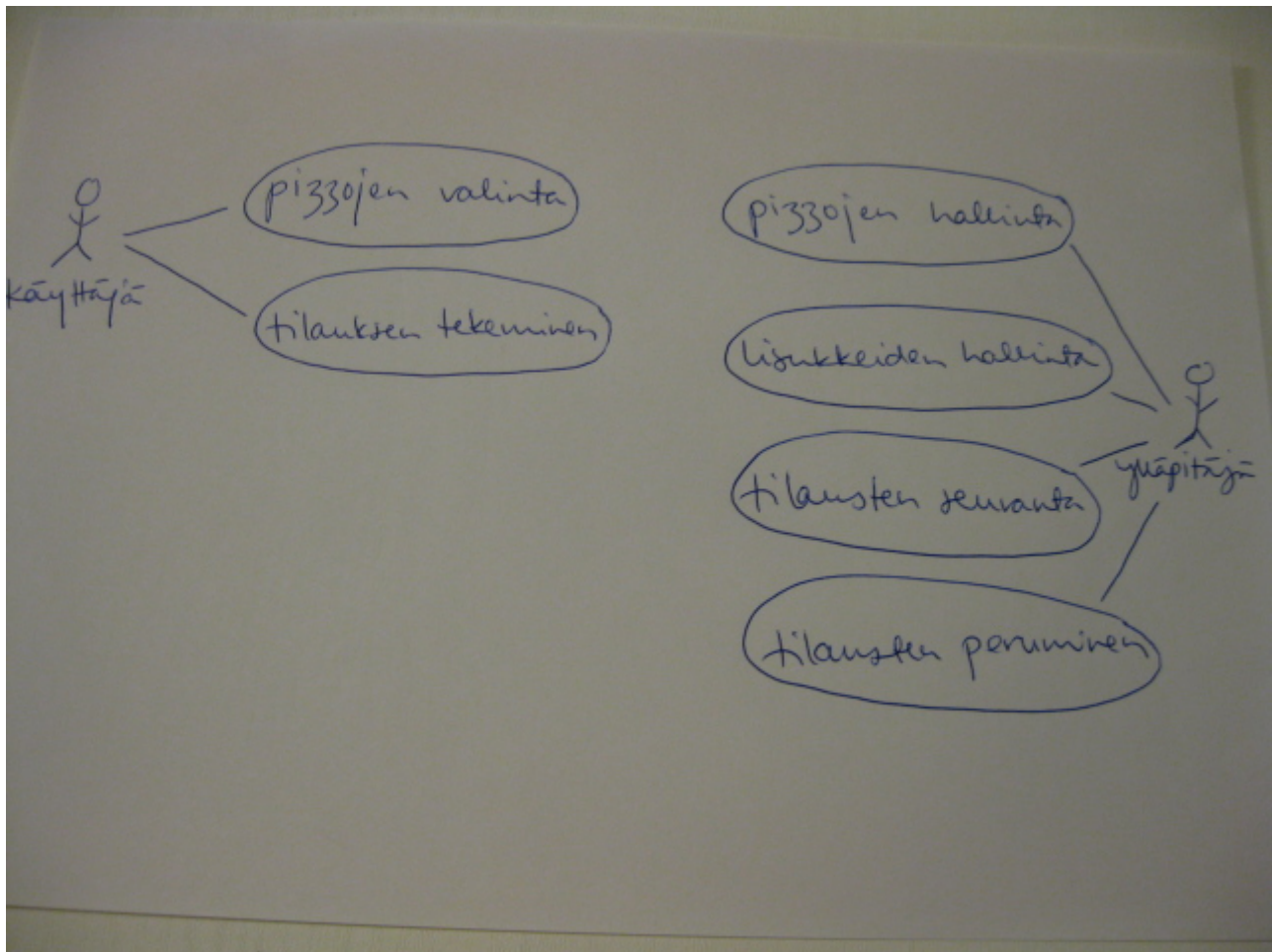
Käyttäjät: ylläpito

Tavoite: hallinnoida tuoteryhmiä

Käyttötapauksen kulku:

Järjestelmässä on eri tuoteryhmiä, esim. ”pizzat” ja ”kebabit”. Sovellus sallii tuotteen lisäyksen tuoteryhmään. Ensimmäisessä versiossa on ainoastaan yksi ”pizzat”-tuoteryhmä.

Käyttötapauskaavio:

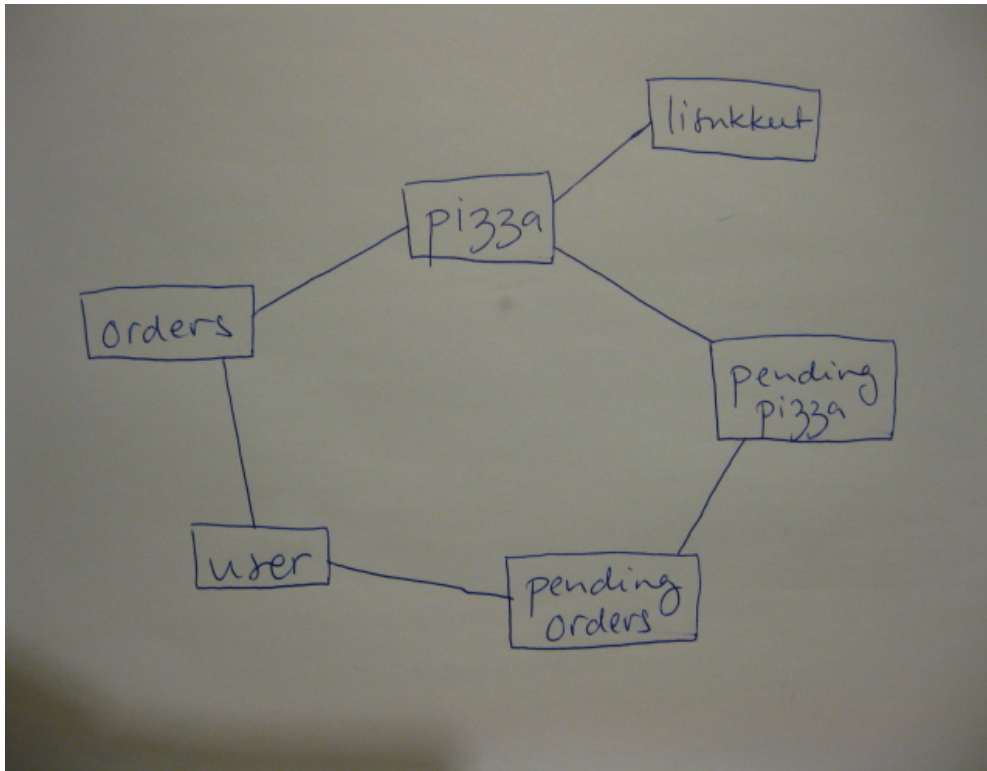


Kayttajaryhmat:

- Anonyymi käyttäjä
 - Anonyymilla tarkoitetaan keta tahansa joka Internet valityksella tulee käyttämään Pizza-Palvelun tilauspalvelun sivuja.
- Ylläpitäjä
 - Ylläpitäjällä tarkotetaan Pizza-Palvelun pizzoja myyvän tahon salasanan takana suojatusta käyttajasta.

Järjestelmän tietosisältö:

Järjestelmä toimii tällä hetkellä kirjoittaessa kahden taulu varassa, ja yhden liitos-taulu kanssa. Taulut **pizzat** ja **lisukkeet** ovat monesta-moneen -relaatio. Liitostaululla esitetään relaatio toteutusmuodossa.



Sovellus toimii näiden kahden taulun varassa. Lisäksi toteutetaan sisäänkirjaantuminen ja omien pizzojen koostaminen lisukkeista. Nämä ovat tärkeitä tapauksia. Ajan antaessa periksi sovelletaan myös muita ominaisuuksia sovellukseen.

Käyttöohjeet:

Sovelluksen etusivulla pääsee syöttämällä selaimen osoitteen <http://brute.havoc.fi:8800/tsoha/>. Sivulla on linkkejä toimintoihin. Anonyyminä käyttäjänä saat näkyviin seuraavat näkymät:

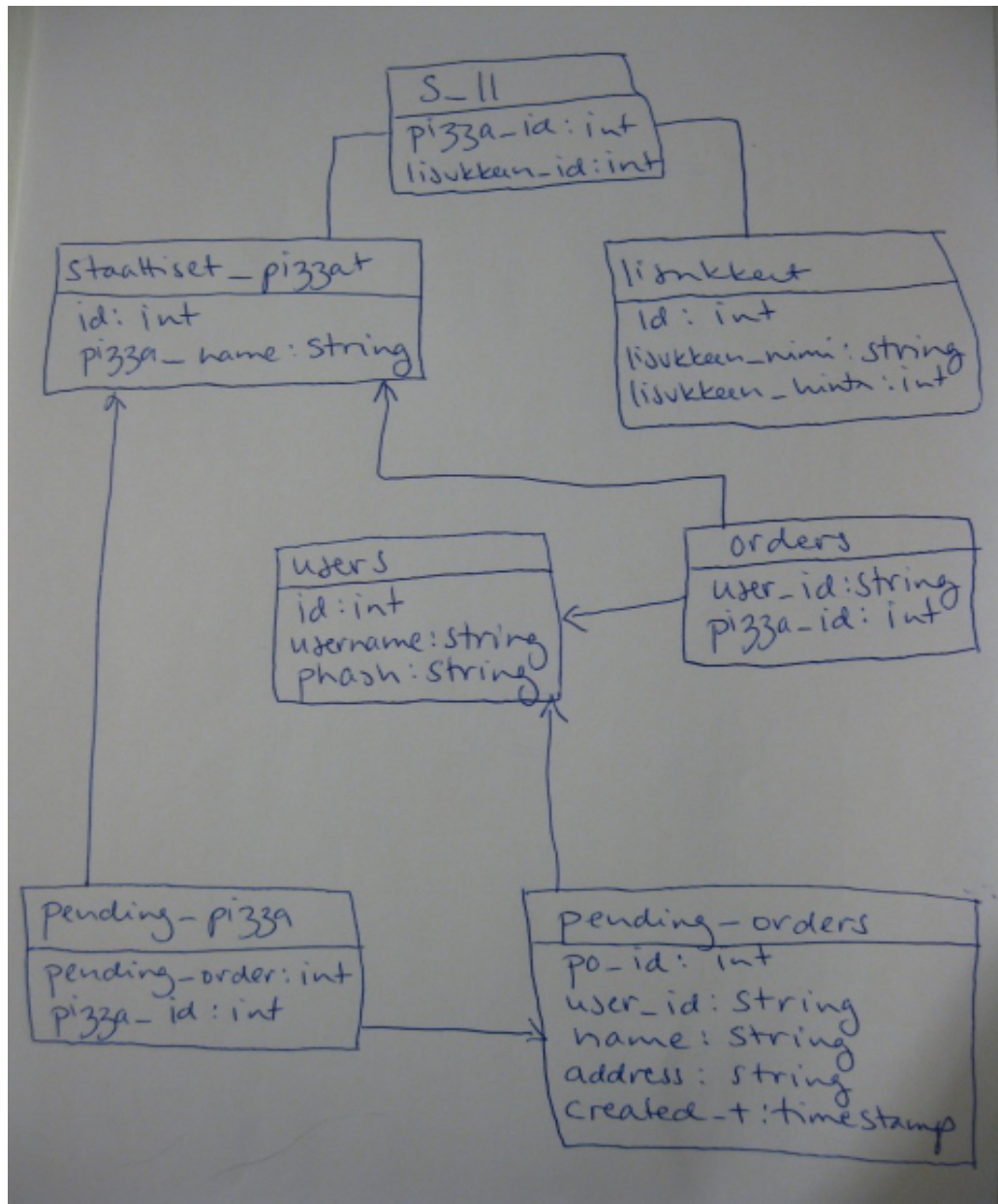
1. Menuun pizzoja jotka ovat valmiiksi määriteltäviä.
2. Voit katsoa tarkempia tietoja pizzasta *lisätiedot* -linkkiä painamalla.

Käyttäjä voi valita pizzoja tilaukseensa painamalla linkkiä pizza:n oheessa. Tilauksensa voi poistaa menemällä menuun kautta *Tilaa pizzat* -sivulle, ja valitsemalla *Perua tilaus*. Pizzat tilataan painamalla samalla sivulla *Tilaa pizzat* -linkkiä, jolla annetaan nimi ja osoite jonne pizzat toimitetaan.

Admin -käyttäjä voi tämän lisäksi:

1. Lisätä lisukkeita.
2. Lisätä valmiiksi määrittelyjä pizzoja.
3. Seurata tilauksia.

Relaatiotietokantakaavio



Jarjestelman yleisrakenne

Sovellus on MVC -mallin mukaisesti tehty sovellus. Periaatteessa sovellus makaa **Slim** -nimisen sovelluskehityksen takana. Seuraavia asioita hoidetaan sovelluskehityksen kautta:

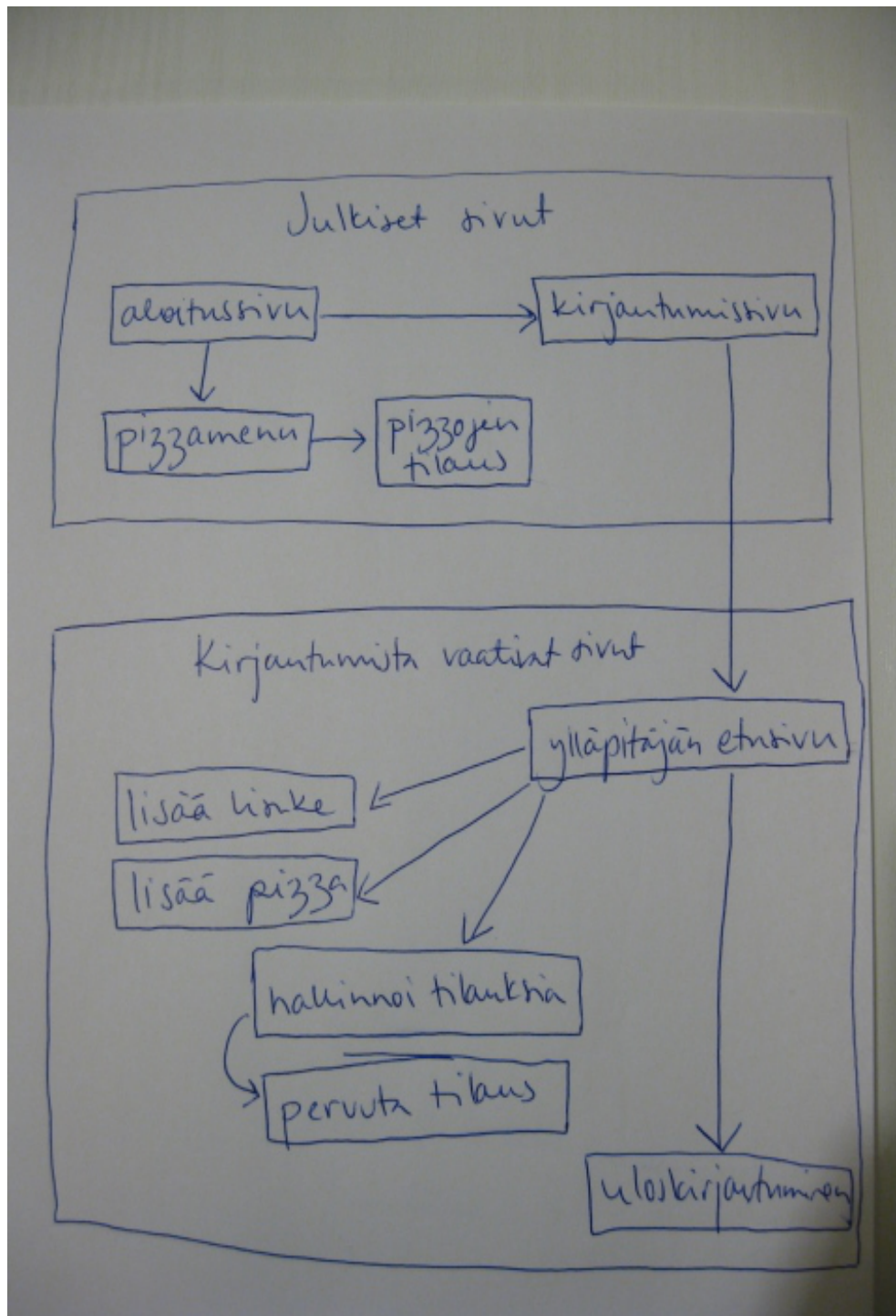
1. Tietokantayhteydet. Toimitetaan automaattisesti *DB::connect()* -metoodin kautta kahva jonka avulla syotetaan kaikki asiat tietokantaa ja tietokannasta. Myös salasana on määritelty tässä kontekstissa.
2. Template -kieli on **Twig** -nimisen ohjelmiston toteuttama käyttöliittymän rakennus-kehys jossa on PHP:n oliomallia tukeva liittyminen. Myös templatien kautta voidaan esittää ns. *blockeja* -koodia.
3. *Kint* toimii debuggerina ongelmatilanteissa.
4. ...

Sovellus toimii mallin kautta haetun tiedon perusteella controllerin kautta template-kieleen. Sovelluksen pääpiirteet ovat `app/{models,controllers,views}` -hakemistojen sisältä. Sovellus toteuttaa olio-rajapinnan `models` -luokkien päälle. Tämä tarkoittaa sitä että sovellus:

1. Syöttää itsensä tietokannasta hakemien tietojen perusteella. Eli luokka populoi itsensä esim., avaimen perusteella tietokannasta, ja muodotettu olio palautetaan takaisin staattisen metoodin kautta ilmentymäksi.
2. Olio voi täyttää itsensä annetuista arvoista, ja työntää arvot tietokantaan. Talloin voidaan helposti antaa esim., `$olio->store()` -kutsu, joka syöttää olion konseptuaalisesti kantaan.
3. Olio-malli toteuttaa avaimen kautta poistamisen tietokannasta.
4. Olio-malli toimii rekursiivisesti, eli esimerkkinä *pizza* voi sisältää *lisukkeita*, joten *Pizza* -malli kutsuu *Lisuke*-mallia ja pyytää siltä listan olioita liitettäväksi *Pizza*-olioon.

Sovelluksen objektimalli todettiin hyvin selkeästi ylivoimaiseksi ns., *ad hoc* -mallia vastaan verrattuna. Sovellus sisältää konseptuaalisesti **Jarjestelman tietosisälto** -osan kohdalla mainituista olioista koostuvan mallin. Sovelluksen alkuvaiheissa hydynettiin vain tiedon palauttamista suoraan ilman olio-mallia näkymälle suoritettavaksi, ja malli jätettiin pois laskuista. Ohjelma muodosti nopeaan tahtiin hankalasti hallittavaksi koska jokainen näkymä vaati oman mallinsa. Malli toteutettiin tietosisällön perusteella ja havaittiin ylivoimaiseksi tavaksi hallinnoida tietoa.

Käyttöliittymä ja järjestelmän komponentit



Asennustiedot

Sovellus on tehty toimimaan Linux pohjaisessa palvelimessa. Sovellus toimii:

1. Apache 2.4.10
2. Slim versio 2
3. mod_php5 5.6.26
4. PostgreSQL 9.4.9

Naiden sovellusten varassa toimiva <https://github.com/Tsoha/Tsoha-Bootstrap> Git -repository pitää huolen myös sisäisesti että sovellus toimii. Apachen konfiguraatiossa on tehty moduulien käynnistys, lahina:

1. mod_php
2. mod_rewrite
3. ...

Asennus ei ole kovin hyvin maariteltu sovelluskehitykseen käytettyjen maarittelmien suhteen. Tarvittavat moduulit asennettuna ja saatettuna oikein, antaa ympariston joka esitetaan *NameVirtualHost* -maarityksen kautta, ja tsoha asennetaan siten että sovelluksen tiedostot tulevat sellaiseen paikkaan että ne saadaan kasiksi verkon kautta osoitteesta `http://...../tsoha/`. Tama johtuu siitä että hirmu useassa paikassa ollaan tehty oletus että sovellus toimii `/tsoha/` -hakemistosta kasin. Yksi tapa tehdä tama toiminnallisuus on tehdä `/var/www/html` -hakemistoon symbolinen linkki esim.:

```
# cd /var/www/html
# ln -s /home/${whoami}/tsoha/deploy tsoha
```

Nain *saadaan* kaytoon `/tsoha` -hakemisto. **mod_rewrite** on tarpeellinen että voidaan siirtää pyyntöjen kasittely suoraan `tsoha/deploy/index.php` -scriptin kasiteltavaksi.

Sovellus vaatii PostgreSQL -tietokannalta tukea toimiakseen. Se miten tietokanta luodaan, ja kuinka tehdään kayttaja siihen on taman dokumentin laajuuden ulkopuolella.

Sovellus vaatii toimiakseen `bootstrap.sh` -tiedoston ajamisen. Ennen kun ajat `bootstrap.sh`:n muista muuttaa `$TSOHA_BASE` ja `$TSOHA_DEPLOY` muuttujia `bootstrap.sh` -tiedostossa osoittamaan sinun *Tsoha-Bootstrap* -hakemistoon ja `deploy` -hakemistoon, esim.:

```
TSOHA_BASE=$HOME/tsoha/Tsoha-Bootstrap
TSOHA_DEPLOY=$HOME/tsoha/deploy
```

Editoi myös `deploy.sh` -tiedoston `$TSOHA_BASE` aj `$TSOHA_DEPLOY` muuttujia samanlaisiksi kuin `bootstrap.sh` -tiedostossa. Tassa vaiheessa on hyva idea kokeilla toimiiko **psql** -ohjelma. Aja:

```
$ psql tietokannan_nimi
```

Miten tietokanta asetetaan toimivaksi ei ole taman dokumentin laajuudessa. Jos jarjestelma on maariteltu oikein pitaisi sinun saada *tietokannan_nimi*=> kehoite terminaaliin. Aja seuraavaksi:

```
$ sh create_tables.sh
```

Se luo sinulle taulut joita sovellus kayttaa. Nyt olet valmis ajamaan `bootstrap.sh` -tiedoston:

```
$ sh bootstrap.sh
```

Ja jos se onnistuu, aja:

```
$ sh deploy.sh
```

Ja tata kautta sovelluksen pitaisi olla kaytettavissa.

Kaynnistys- / kayttoohje

Jos seurasit tarkasti **asennustiedot** -kategorian ohjeita, pitäisi sinulla olla katyettava Pizza-palvelu osoitteessa <http://sinunpalvelimen-osoite/tsoha>.

Kehitetty sovellus toimii osoitteessa <http://brute.havoc.fi:8800/tsoha> ja järjestelmaa voi testata käyttäjätunnuksella ja salasanalla:

Kayttaja 1:

Kayttajatunnus: admin

Salasana: foobar

Testaus, tunnetut bugit, puuttet & jatkokehitysideat

Sovellusta ei ole kategorisesti testattu. Sovelluksen laajuuden supistumisen myötä myös sovelluksen mahdolliset bugit ovat aika hyvin hallittuja. Sovelluksessa *tilausten seuranta* -sivulla näkyy liian suppeasti pizzat jotka liittyvät tilaukseen. Sovelluksen puutteita ovat tehottomuus. Sovellus tekee liikaa kyselyitä hakeakseen esim. Pizza -mallin kanssa Lisukke-mallin olioita yksitellen, eli jos lisukkeita on monta, ja pizzoja on monta tehdään sovelluksessa hurja määrä erillisiä SQL -kyselyitä. Sovellus ajetaan **armv7** -pohjaisella palvelimella jossa tehottomuus tulee heti selville kun kannassa on muutama pizza ja lisuke.

Jatkokehitys-ideoita on paljon. Toimiva järjestelmä voisi tukea käyttäjä-tunnuksia tilaajille. Näin saataisiin esim., toimitus -virheet kirjattua kantaan, ja olisi helpompaa peruuttaa tilauksia jotka tulevat ongelmallisista osoitteista.

Omat kokemukset

Sovelluksen kattavuus on huikean iso jos sita siirtyisi tekemaan valmiiksi tuoteeksi pizzerioille. Testaus-mielessa sovellus tukee periaatteellisia asioita kuten pizzojen tilauksia ja tilauksten hallinnointia. Oikea sovellus sisältää huiman paljon erillaisia ominaisuuksia. Lahdin tekemaan harjoitustyota naivista lahtokohdasta, olettaen etta järjestelmä olisi helppo ja nopea ratkaista. Sovelluksen kehityksen aikana palautui mieleen Ohjelmistotekniikan menetelmät -kurssin opetukset, ja iteroin tiedon suorasta syöttämisestä mallista näkymään, mikä osoittautui harvinaisen hankalaksi tavaksi hallinnoida tietoa. Siirtyminen olio-malliin helpotti tilannetta huomattavasti, ja osoittautui etta tietyt toiminnallisuudet sopivat hyvin yhteen selkeän mallin kanssa.

Sovelluksen kehityksessä on tietoisesti oltu käyttämättä Javascript -koodia verkko-sivuilla.