

Tietokantasovellus loppukesä 2011  
Toteutusdokumentti  
Pitsapalvelu  
Santtu Keskinen

# Sisällysluettelo

1. Johdanto.....	1
2. Ohjelmiston yleisrakenne.....	1
3. Sivujen kuvaukset.....	2
Julkisen osan sivut.....	2
Admin osan sivut.....	4
4. Asennustiedot.....	5
5. Käynnistys- ja käyttöohje.....	5
6. Liitteet.....	7

## 1. Johdanto

Pitsapalvelu on web-sovellus joka palvelee pitseriaa ja sen asiakkaita. Järjestelmään kuuluu selaimella käytettävä käyttöliittymä ja tietokanta. Pitseria voi sovelluksen avulla määritellä hinnastot ja asiakkaat tehdä tilauksia. Muita käyttökohteita on muunmuassa tilausten ja niiden toimitusten seurauksen seuranta ja tulostus.

Hyvän pitsapalvelun ominaisuuksiin kuuluu tilaamisen helppous. Pitserialle tuotetietojen muokkaamisen tarvitsee olla riittävän yksinkertaista, mutta monipuolista. Pitserian pitää nähdä ja pystyä tulostamaan lista odottavista tilauksista. Tilausten tilat voi toimituksen jälkeen muuttaa toimitetuiksi ja tilausten kuitteja täytyy pystyä tulostamaan.

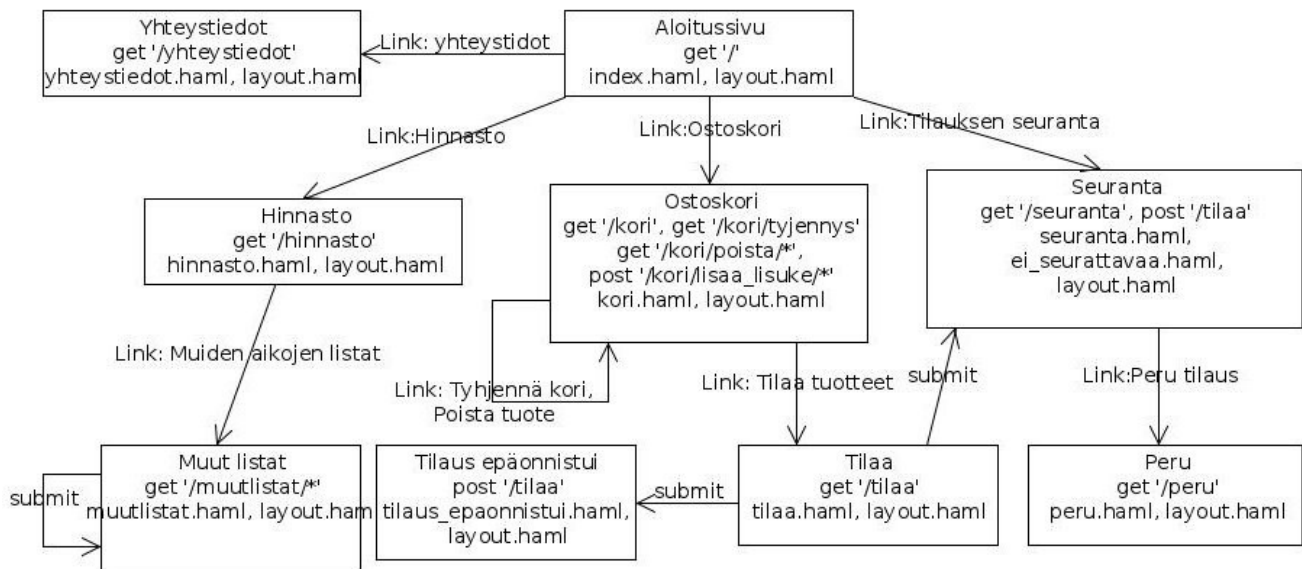
Ominaisuudet on toteutettu suunnitteludokumentin mukaisina lukuunottamatta asiakkaan mahdollisuutta muuttaa täyhteitä itse joissain tuotteissa, jota ei ole toteutettu.

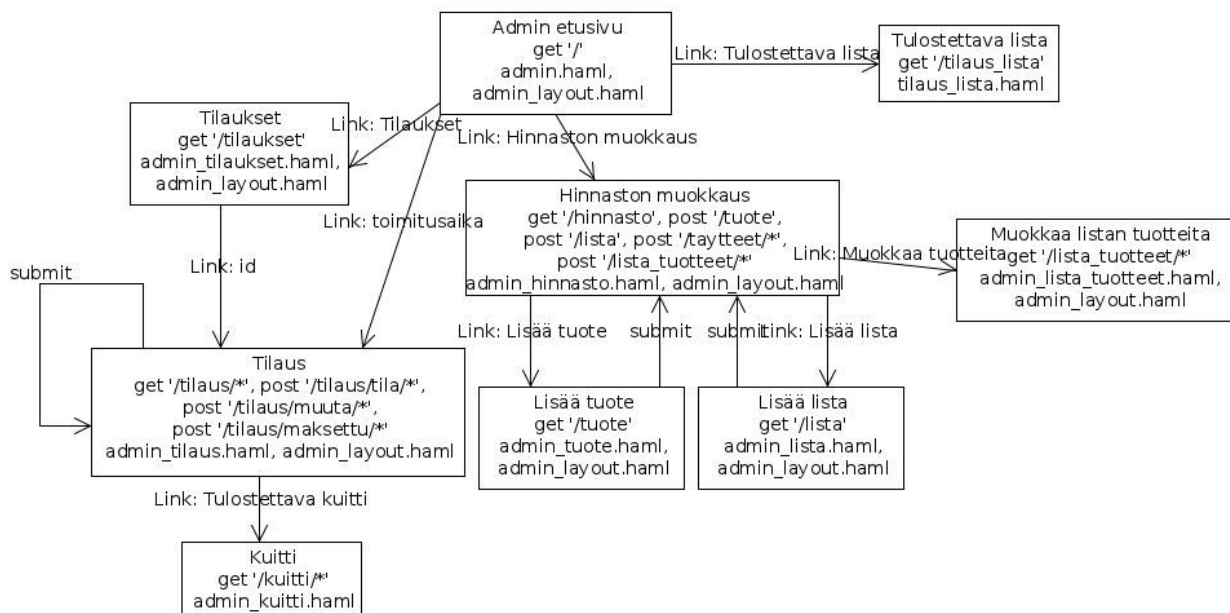
## 2. Ohjelmiston yleisrakenne

Jokainen sivu koostuu parista ohjain(controller) ja näkymät(views). Kaikki sivut ovat siis generoituja vaikka niissä olisikin staattinen sisältö.

Rakennekaavioissa on kerrottu kuhunkin sivuun liittyvät ohjaimet ja näkymät sekä sivun nimi.

Erikseen on annettu rakennekaaviot julkiselle osuudelle ja vain pitserian käytössä olevalle osuudelle.





### 3. Sivujen kuvaukset

#### Julkisen osan sivut

##### Aloitussivu:

Tyhjä alkusivu.

Ohjaimet: get '/'

Näkymät: index.haml, layout.haml

Linkit: Yhteystiedot, Hinnasto, Ostoskori, Seuranta

##### Yhteystiedot:

Tyhjä yhteystietosivu

Ohjaimet: get '/yhteystiedot'

Näkymät: yhteystiedot.haml, layout.haml

##### Hinnasto:

Sivu jolla on kuvaus tämänhetkisestä ruokalistasta ja sen sisältämät tuotteet. Sivulla voi lisätä tuotteita ostoskoriin.

Ohjaimet: get '/hinnasto'

Näkymät: hinnasto.haml, layout.haml

Javascript: lisää tuote ostoskoriin

**Muut listat:**

Sivu jolla voi katsoa kaikkia ruokalistoja riippumatta siitä mikä on tällähetkellä voimassa.

Ohjaimet: `get /muutlistat/*`

Näkymät: `muutlistat.haml`, `layout.haml`

Lomake: Valitsee mitä listaa näytetään. Päätyy: Muut listat

**Ostoskori:**

Sivu jolta näkee mitä ostoskorissa on. Lisäksi sivulla voi lisätä lisukkeita tuotteisiin ja poistaa tuotteita tai tyhjentää koko ostoskorin.

Ohjaimet: `get '/kori'`, `get '/kori/tyhjennys'`, `get '/kori/poista/*'`, `post '/kori/lisaa_lisuke/*'`

Näkymät: `kori.haml`, `layout.haml`

Linkit: Ostoskori, Tilaa

**Tilaa:**

Sivu jolla syötetään tiedot tilausta varten.

Ohjaimet: `get '/tilaa'`

Näkymät: `tilaa.haml`, `layout.haml`

Lomake: Syötettävät tiedot tilausta varten. Päätyy: Tilaus epäonnistui, Seuranta

**Tilaus epäonnistui:**

Sivu jolle joutuu jos tilatessa syötetyt tiedot eivät kelvanneet.

Ohjaimet: `post '/tilaa'`

Näkymät: `tilaus_epaonnistui.haml`, `layout.haml`

**Seuranta:**

Sivu jolta näkee tietoa tehdystä tilauksesta ja voi perua tilauksen.

Ohjaimet: `get '/seuranta'`, `post '/tilaa'`

Näkymät: `seuranta.haml`, `ei_seurattavaa.haml`, `layout.haml`

Linkit: Peru

**Peru:**

Sivu jolle joutuu jos peruu tai yrittää perua tilauksen.

Ohjaimet: `get '/peru'`

Näkymät: `peru.haml`, `layout.haml`

## **Admin osan sivut**

Admin osaan pääsee lisäämällä osoitteeseen polun /admin/. Esimerkiksi /tilaus -> /admin/tilaus.

### **Admin etusivu:**

Admin puolen etusivu jolla näkee odottavat tilaukset.

Ohjaimet: get '/'

Näkymät: admin.haml, admin\_layout.haml

Linkit: Tilaukset, Hinnaston muokkaus, Tulostettava lista, Tilaus

### **Tilaukset:**

Lista kaikista tilauksista kaikilta ajoilta.

Ohjaimet: get '/tilaukset'

Näkymät: admin\_tilaukset.haml, admin\_layout.haml

Linkit: Tilaus

### **Tilaus:**

Yksittäisen tilauksen sivu. Tältä sivulta voi muokata tilauksen tilaa.

Ohjaimet: get '/tilaus/\*', post '/tilaus/tila/\*', post '/tilaus/muuta/\*', post '/tilaus/maksettu/\*'

Näkymät: admin\_tilaus.haml, admin\_layout.haml

Linkit: Kuitti

Lomake: Tilauksen muuttamiseen käytettävä lomake. Päätyy: Lomake

### **Kuitti:**

Tulostetta kuitti jonka voi tulosta tilaukselle.

Ohjaimet: get '/kuitti/\*'

Näkymät: admin\_kuitti.haml

### **Tulostettava Lista:**

Tulostettava lista odottavista tilauksista.

Ohjaimet: get '/tilaus\_lista'

Näkymät: tilaus\_lista.haml

### **Hinnaston muokkaus:**

Sivu jolta voi muokata hinnastoa eli täytteitä, listoja ja tuotteita.

Ohjaimet: get '/hinnasto', post '/tuote', post '/lista', post '/täytteet/\*', post '/lista\_tuotteet/\*'

Näkymät: admin\_hinnasto.haml

### **Lisää tuote:**

Sivu jolla voi lisätä uuden tuotteen.

Ohjaimet: `get '/tuote'`

Näkymät: `admin_tuote.html`, `admin_layout.html`

Lomake: Uuden tuotteen tietoja. Päätyy: Hinnaston muokkaus

### **Lisää lista:**

Sivu jolla voi lisätä uuden ruokalistan.

Ohjaimet: `get '/lista'`

Näkymät: `admin_lista.html`, `admin_layout.html`

Lomake: Uuden listan tietoja. Päätyy: Hinnaston muokkaus

### **Muokkaa listan tuotteita:**

Sivu jolla voi muokata ruokalistaan sisältyviä tuotteita ja niiden hintoja.

Ohjaimet: `get '/lista_tuotteet/*'`

Näkymät: `admin_lista_tuotteet.html`, `admin_layout.html`

Lomake: Listaan kuuluvat tuotteet ja niiden hinnat. Päätyy: Hinnaston muokkaus

## **4. Asennustiedot**

Sovellus käyttää Sinatra-kehystä. Suurin osa toimintaan liittyvistä tiedostoista sijaitsee sovelluksen juurihakemistossa. Juuressa on `config.ru` josta suoritus lähtee liikkeelle sekä `pitsapalvelu_julkinen.rb` ja `pitsapalvelu_admin.rb` jotka sisältävät reitityksen ja ohjaimet. Lisäksi juuressa on Gemfile joka määrittelee kirjastot josta sovellus on riippuvainen ja `database.db` joka on SQLite tietokanta jota sovellus käyttää.

Näkymät on sijoitettu views hakemistoon. Lähes jokainen generoitu sivu käyttää `layout.html` tai `admin_layout.html` näkymiä. Jokaiselle generoidulle sivulle on lisäksi vähintään yksi oma näkymä.

Staattinen sisältö on sijoitettu public hakemistoon. Hakemisto sisältää javascript-tiedostot ja tyyliarit.

Lisäksi juuressa on doc hakemisto joka sisältää dokumentaatioita ja on merkityksetön sovelluksen toiminnan kannalta.

## **5. Käynnistys- ja käyttöohje**

Sovellusta pyörittävällä konella täytyy olla asennettuna ruby ja tarvittavat *gemit* eli kirjastot. Tarvittavat gemit on määritelty Gemfile tiedostossa. Olettaen että rubyn oma pakentinhallinta on käytettävissä saa tarvittavat gemit asennettua ajamalla sovelluksen juuressa komento *bundle install*.

Sovellus pitäisi olla mahdollista saada toimimaan millä tahansa http-serverillä joka tukee rack-sovelluksia. Yksi mahdollisuus on ajaa sovelluksen juuressa komento *rackup* joka käynnistää rack-serverin (vakiona porttiin 9292). Tämän jälkeen sovellus on käytettävissä osoitteessa `http://localhost:9292`.

Pitserian omaan käyttöön tarkoitettu osio on erotettu muusta sovelluksesta. Tämä osuus on polun /admin/ takana. Todennukseen käytetään yksinkertaista html-todennusta jonka käyttäjä ja salasa ovat molemmat admin.



## 6. Liitteet

Lopulliset käytetyt SQLiten näkymä ja create table lauseet.

```
create view tuote_hinnat as select
lj.tuote_id as tuote,
rl.id as lista,
lj.saatavilla as saatavilla,
lj.hinta as hinta,
rl.alku as alku,
rl.loppu as loppu
from lista_jasen lj, ruokalista rl
where lj.ruokalista_id=rl.id
order by rl.prioriteetti desc;

create table tayte (
id integer not null primary key,
nimi varchar(50) not null
);

create table tayte_tuote (
tayte_id integer not null,
tuote_id integer not null,
primary key (tayte_id, tuote_id),
foreign key (tayte_id) references tayte
on delete restrict on update restrict,
foreign key (tuote_id) references tuote
on delete cascade on update cascade
);

create table tuotetyyppi (
id integer not null primary key,
nimi varchar(50) not null,
taytteet boolean default false,
lisukkeet boolean default false
);
```

```
create table tuote (  
id integer not null primary key,  
tuotetyyppi_id integer,  
nimi varchar(50) not null,  
kuvaus varchar(100),  
foreign key(tuotetyyppi_id) references tuotetyyppi  
on delete cascade on update cascade  
);
```

```
create table tilaus (  
id integer not null primary key,  
maksettu decimal,  
nimi varchar(100),  
puhelin varchar(30),  
osoite varchar(200),  
kotiinkuljetus boolean,  
tilausaika timestamp,  
toimitusaika time,  
tila integer,  
lisatiedot varchar(200),  
muuta varchar(200),  
foreign key (tila) references tila  
on delete set null on update cascade  
);
```

```
create table tilat (  
id integer not null primary key,  
kuvaus varchar(30)  
);
```

```
create table ostos (  
id integer primary key,  
tilaus_id integer not null,  
tuote_id integer,  
ruokalista_id integer,
```

```

    hinta decimal,
    foreign key (tilaus_id) references tilaus
    on delete cascade on update cascade,
    foreign key (tuote_id) references tuote
    on delete set null on update cascade,
    foreign key (ruokalista_id) references ruokalista
    on delete set null on update cascade
);

create table lista_jasen (
    ruokalista_id integer not null,
    tuote_id integer not null,
    saatavilla boolean,
    hinta decimal,
    primary key (ruokalista_id, tuote_id),
    foreign key (ruokalista_id) references ruokalista
    on delete cascade on update cascade,
    foreign key (tuote_id) references tuote
    on delete cascade on update cascade
);

create table ruokalista (
    id integer not null primary key,
    nimi varchar(30),
    alku time,
    loppu time,
    prioriteetti integer,
    kuvaus varchar(50)
);

create table lisuke (
    id integer not null primary key,
    nimi varchar(50)
);

create table lisuke_ostos (

```

```
lisuke_id integer not null,  
ostos_id integer not null,  
primary key (lisuke_id, ostos_id)  
foreign key (lisuke_id) references lisuke  
on delete set null on update cascade,  
foreign key(ostos_id) references ostos  
on delete cascade on update cascade  
);
```