

4.5.2017

Dokumentaatio: Pizzapalvelun tietokantasovellus nimeltään Gizza

Johdanto

Gizza on kuvitteelliselle Pizzeria Omertalle suunniteltu ja toteutettu tietojärjestelmä, jonka pääasiallinen tarkoitus on mahdollistaa pizzojen tilaus Internetin välityksellä. Asiakas siis käyttää Gizzaa selaimella. Hän voi valita tilaukseensa pizzojen lisäksi myös muita tuotteita kuten esim. juomia. Asiakkaan tekemään tilaukseen kuuluu myös tieto siitä, milloin ja mihin osoitteeseen pizzatilaus on tarkoitus toimittaa.

Gizzan ominaisuuksiin kuuluu asiakkaan identifiointi ja asiakkaaseen liittyvien historiatietojen tallentaminen. Käytännössä tämä tarkoittaa sitä, että asiakkaan kannalta järjestelmän käyttäminen edellyttää asiakastilin luomista. Asiakkaan tekemät tilaukset tallennetaan järjestelmään. Sekä asiakas itse että pizzerian henkilökunta voivat myöhemmin tarkastella näitä tilaustietoja. Henkilökunnalla on myös mahdollisuus tallentaa ja tarkastella tietoja siitä, sujuiko tilauksen toimitus kuten piti vai ilmeni ongelmia. Jokainen ongelma liittyy tiettyyn tilaukseen, ja toisaalta jokainen tilaus liittyy tiettyyn asiakkaaseen. Henkilökunta voi siis Gizzan avulla saada tiedon asiakkaista, joiden kanssa on ilmennyt ongelmia. (Gizzan tietokantaan tallennetut asiakkaat ovat puhtaasti kuvitteellisia henkilöitä, joten tässä yhteydessä ei ole tarpeen pohtia henkilötietolain asettamia rajoituksia.)

Gizzan toiminnallisuus määräytyy melko pitkälti sen mukaan, kirjaudutaanko sisään ylläpitäjänä (eli henkilökunnan jäsenenä) vai asiakkaana. Ylläpitäjä voi tarkastella ja muokata kaikkiin tilauksiin, asiakkaisiin ja asiakkaiden osoitekirjoihin liittyviä tietoja. Ylläpitäjä voi myös tehdä, muokata ja poistaa tilauksia asiakkaiden puolesta. Tämä mahdollistaa esim. puhelinasiakkaiden helpon palvelemisen Gizzan avulla.

Asiakas voi Gizzan avulla tehdä, muokata, katsella ja poistaa omia pizzatilauksiaan. Hän voi myös katsella ja muuttaa oman asiakastilinsä tietoja sekä halutessaan poistaa tilinsä. Kenellä tahansa on mahdollisuus luoda Gizzaan uusi asiakastili. Valittava rajoitus asiakkaan kannalta on se, että hän ei itse voi lisätä osoitekirjaansa uusia osoitteita, vaan hänen täytyy pyytää ylläpitäjää tekemään se puolestaan (en ehtinyt toteuttaa kovin toiminnallista osoitekirjaa).

Olen pyrkinyt Gizzassa siihen, että käyttöoikeuksiin liittyvän tietoturvan perusasiat ovat kunnossa. Tavallisen asiakastilin kautta ei siis pitäisi olla mahdollista katsoa tai muuttaa toisiin asiakkaisiin tai heidän tilauksiinsa liittyviä tietoja. Vahva tietoturva ei kuitenkaan ole tämän kurssin kaikkein keskeisin aihe, joten en ole varmistanut tietoturvan toimivuutta millään kovin systemaattisella tavalla.

Gizza on web-sovellus, jonka alustajärjestelmän on tuettava PHP-ohjelmointikieltä ja PostgreSQL-tietokantaa. Oletus on, että sovellus ei tule olemaan helposti siirrettävissä (portattavissa) eri tietokantajärjestelmien välillä. Käyttäjän selaimelta ei vaadita erityistä tukea tietylle ohjelmointikielelle. Gizzan toteutus- ja toimintaympäristö on palvelin `users.cs.helsinki.fi` Apache-palvelun alla. Palvelimen users PHP-versio on 5.3.2-1ubuntu4.30 ja PostgreSQL-versio on 8.4.22.

Käyttötapaukset

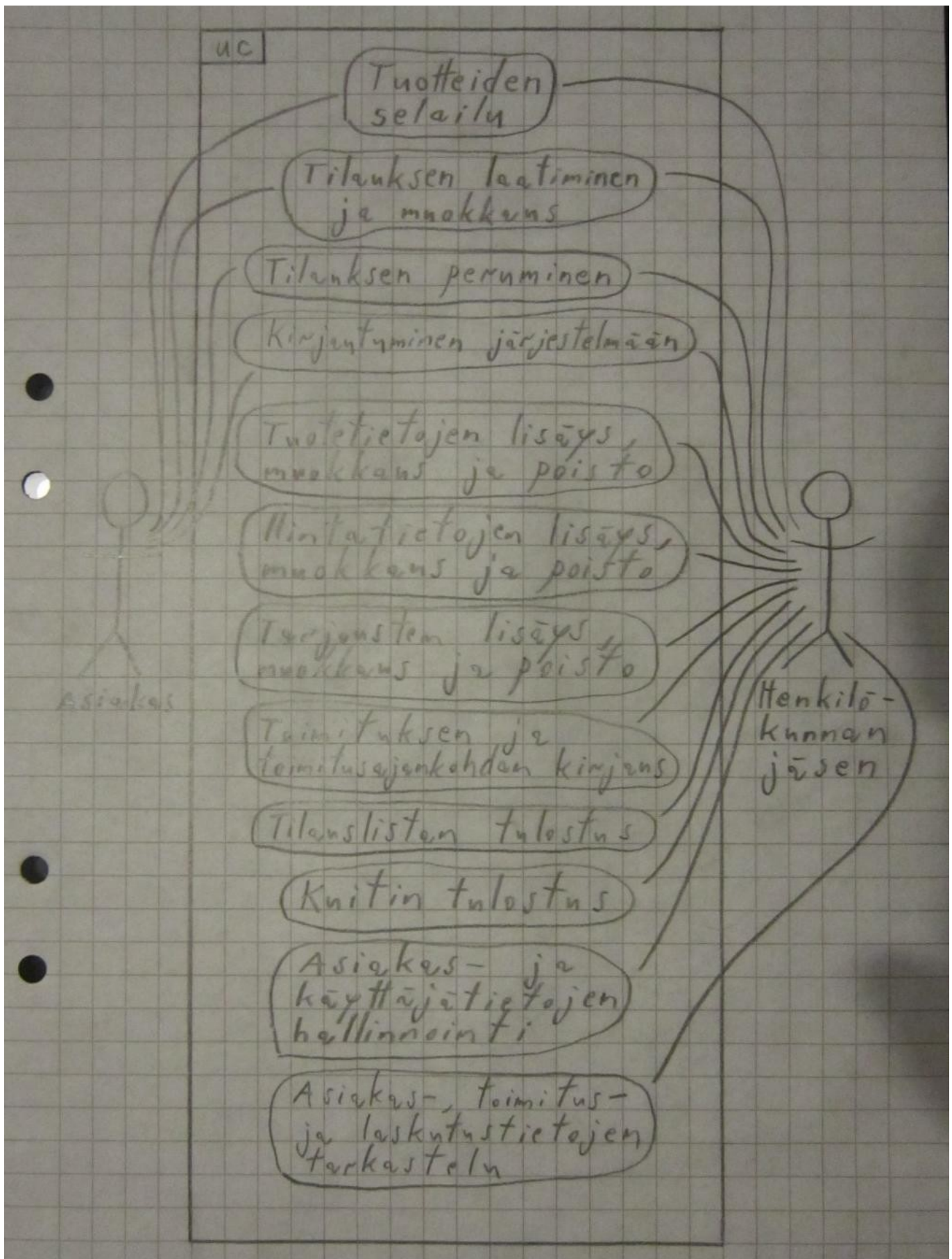
Gizzan sidosryhminä ovat lähinnä asiakkaat ja henkilökunnan jäsenet. Ainakin jälkimmäinen ryhmä voitaisiin periaatteessa jakaa edelleen pienempiin osiin,

4.5.2017

kuten esim. johtoon, tilausten valmistajiin ja kuljettajiin. Tässä vaiheessa tuntuu kuitenkin järkevimmältä hahmottaa sidosryhmiksi vain asiakkaat ja henkilökunnan jäsenet. Asiakas on siis henkilö, joka tekee pizzeriaan tilauksen Gizzan välityksellä. Henkilökunnan jäsenet ovat tietysti pizzerian työntekijöitä.

Seuraavalta sivulta löytyy Gizzan käyttötapauskaavion ensimmäinen versio. Siinä luetellut käyttötapaukset tuntuvat varsin helposti hahmotettavilta, joten kuvailen ne sanalliset vain lyhyesti.

4.5.2017



4.5.2017

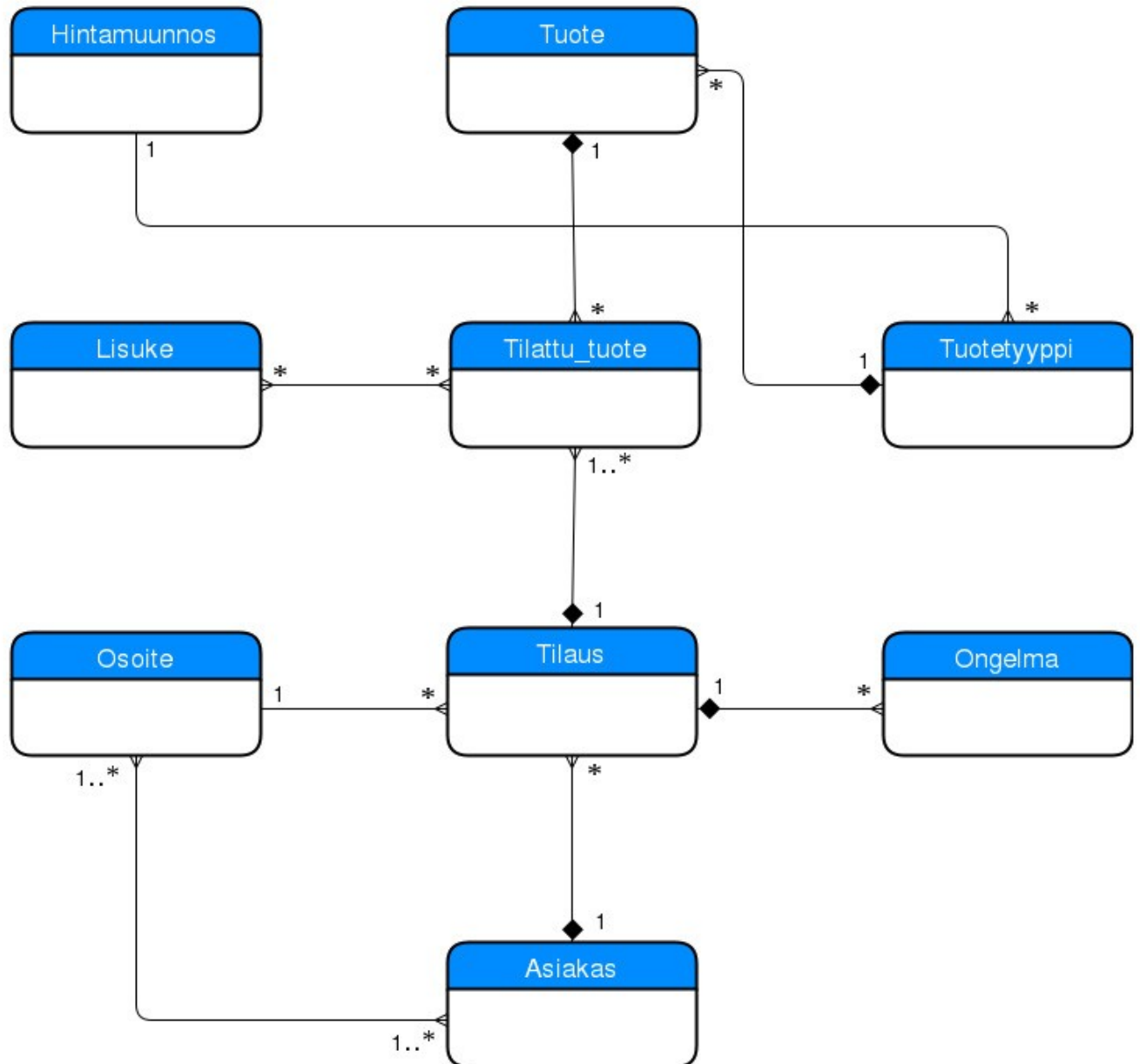
Seuraavassa taulukossa alleviivatut käyttötapaukset tarkoittavat niitä käyttötapauksia, jotka liittyvät sekä asiakkaisiin että henkilökunnan jäseniin. Huomaa, että kaikki käyttötapaukset liittyvät henkilökunnan jäseniin, mutta vain neljä käyttötapausta liittyy asiakkaisiin.

Käyttötapaus	Kommentti
<u>Tuotteiden selailu</u>	Sekä henkilökunta että asiakkaat voivat selailla pizzerian tuotevalikoimaa
<u>Tilauksen laatiminen ja muokkaus</u>	Asiakas voi muokata tilausta, jos sovittuun toimitusaikaan on yli tunti. Tuntuu järkevältä, että myös henkilökunnan jäsen voi laatia tilauksen esim. puhelinasiakkaan puolesta.
<u>Tilauksen peruminen</u>	Asiakas voi perua tilauksen, jos sovittuun toimitusaikaan on yli tunti
<u>Kirjautuminen järjestelmään</u>	Molemmat sidosryhmät tarvitsevat järjestelmän käyttöön käyttäjätunnuksen ja salasanan
Tuotetietojen lisäys, muokkaus ja poisto	Järjestelmä voisi automaattisesti pitää kirjaa siitä, paljonko tiettyä tuotetta (tai raaka-ainetta) on vielä jäljellä
Hintatietojen lisäys, muokkaus ja poisto	Hintatietoihin kuuluu myös tieto siitä, miten esim. vuorokaudenaika vaikuttaa hintaan
Tarjousten lisäys, muokkaus ja poisto	Tarjouksiin voi liittää myös ajankohdan, jolloin ne raukeavat. Tällöin tarjous ei enää näy asiakkaille.
Toimituksen ja toimitusajankohdan kirjaus	Toivotun ja todellisen toimitusajankohdan avulla järjestelmä voi laskea asiakkaan saaman myöhästymisalennuksen.
Tilauslistan tulostus	–
Kuitin tulostus	–
Asiakas- ja käyttäjätietojen tarkastelu sekä hallinnointi	Järjestelmällä voisi olla yksi käyttäjä ylläpitäjän valtuuksilla. Hän voisi jakaa muille henkilökunnan jäsenille näiden tarvitsemat oikeudet järjestelmän käyttöön.
Tilaus-, toimitus- ja laskutustietojen tarkastelu sekä hallinnointi	–

4.5.2017

Järjestelmän tietosisältö

Seuraavassa on järjestelmän keskeinen tietosisältö kuvattu käsitekaavion avulla.



Ennen relaatiotietokantakaavion luomista käydään tietokohteet läpi tarkemmin kuvailemalla niiden tarkoitusta ja keskeisiä attribuutteja.

Tietokohde: Asiakas

Asiakas on Gizzassa tilausten tekijä ja hän on toisaalta se, jolle tilaukset toimitetaan ja joka ne maksaa. Perusidea on, että jokaisella asiakkaalla on käyttäjätunnus Gizzassa. Asiakas-tietokohteeseen kuuluu käsitteellisesti

4.5.2017

jonkinlainen osoitekirja. Asiakas- ja osoitetaulujen välillä on monesta moneen -suhde, joka on toteutettu välitaulun avulla.

Tietokohde Asiakas sisältää tosiasiassa kaikki Gizzan käyttäjätunnukset eli myös ne, jotka kuuluvat henkilökunnan jäsenille. Asiakkaat ja pääkäyttäjät (henkilökunnan jäsenet) erottaa toisistaan attribuutti `on_paakayttaja`, jonka arvo on `true`, jos kyseessä on pääkäyttäjä.

Attribuutti	Arvojoukko	Kuvaus
<u>ktunnus</u>	Merkkijono. Saa sisältää vain pieniä kirjaimia. Pituus vähintään 3 ja enintään 20 merkkiä.	Pääavain. Käyttäjätunnus, jolla asiakas tai henkilökunnan jäsen kirjautuu Gizzaan.
on_paakayttaja	Boolean	Henkilökunnan jäsenet ovat pääkäyttäjiä.
Salasana	Merkkijono. Null-arvo ilmaisee, että käyttäjätunnus on poistettu (väliaikaisesti) käytöstä.	
etunimi	Merkkijono	
sukunimi	Merkkijono	
puhelinnumero	Merkkijono, mieluiten säännöllisen lausekkeen mukainen	
sahkopostiosoite	Kuten yllä	

Tietokohde: Osoite

Täsmälleen sama osoite (esim. Satukatu 1 A 2, 12345 Satukylä) ei saisi toistua taulussa. PostgreSQL:llä tällaisen toistumisen voi ilmeisesti estää `create table` -rakenteen rivillä `UNIQUE(lahiosoite, postinumero, postitoimipaikka)`.

Attribuutti	Arvojoukko	Kuvaus
<u>osoite_id</u>	int	Pääavain, joka toimii muissa tauluissa kompaktina viiteavaimena
lahiosoite	Merkkijono	
postinumero	Merkkijono	
postitoimipaikka	Merkkijono	

4.5.2017

Tietokohde: mm_Asiakas_Osoite

Tällä taululla toteutetaan monesta moneen -yhteys Asiakas- ja Osoite-taulujen välillä. Pääavain on siis kaksiosainen.

Attribuutti	Arvojoukko	Kuvaus
<u>ktunnus</u>	string	Toimii myös viiteavaimena Asiakas-tilaukseen
<u>osoite_id</u>	int	Toimii myös viiteavaimena Osoite-tilaukseen

Tietokohde: Tilaus

Tilaus- ja Osoite-tilaukseen välillä on suora yhteys. Tämä voi ensikatsomalta vaikuttaa turhalta, koska myös Asiakas-tilaukseen on suora yhteys. Asiakkaalla voi kuitenkin olla useampi kuin yksi osoite. Oletetaan, että jos tilaus on luovutettu asiakkaalle (eli kentän ts_tak_toteutunut arvo ei ole NULL), niin asiakas on myös maksanut tilauksen. Attribuutit ktunnus ja ts_tilauksen_teko on merkitty **vahvennetulla**. Tämä tarkoittaa sitä, että parin (ktunnus, ts_tilauksen_teko) on tarkoitus olla taulussa uniikki (unique constraint). Tämä ilmentää sitä, että tilauksen olemassaolo on riippuvainen sen tehneestä asiakkaasta.

Attribuutti	Arvojoukko	Kuvaus
<u>tilaus_id</u>	int	Pääavain
ktunnus	string	Viiteavain Asiakas-tilaukseen
ts_tilauksen_teko	Timestamp	Hetki, jona tilaus on jätetty järjestelmään
ts_tak_toivottu	Timestamp	Tak eli toimitusajankohta. Asiakkaan toivoma ajankohta toimitukselle. Arvo NULL tarkoittaa "mahdollisimman pian".
ts_tak_toteutunut	Timestamp	Ajankohta, jolloin toimitus luovutettiin sen maksaneelle asiakkaalle. Ennen luovutusta arvo on NULL.
osoite_id	int	Viiteavain Osoite-tilaukseen

Tietokohde: Ongelma

Tiettyyn tilaukseen liittyy nolla tai useampi ongelmaa, ja toisaalta jokainen ongelma liittyy johonkin tiettyyn tilaukseen. Ongelma-tietokohteen avulla on tarkoitus tallentaa järjestelmään tietoja lähinnä asiakkaiden kanssa tulleista

4.5.2017

ongelmista. Esim. tilaukseen 1234 voi liittyä ongelmat "tilauksen toimittaja pahoinpideltiin" ja "asiakas kieltäytyi maksamasta tilausta vaikka otti sen vastaan". Tuntuu järkevältä rajata ongelmien päätyypit esim. kolmeen: (1) väkivaltainen asiakas, (2) asiakasta ei tavoitettu, (3) asiakas ei ollut maksukykyinen. Varsinaisessa PostgreSQL-toteutuksessa tätä voisi vastata seuraava enum: `create type Ongelma_enum as enum('violence', 'customer_not_found', 'no_payment');`

Huomaa kaksiosainen pääavain. Avaimen ensimmäinen sarake yksilöi tilauksen. Toinen sarake on kolmiarvoinen enum, joten yksittäiseen tilaukseen voi liittyä korkeintaan kolme ongelmaa.

Attribuutti	Arvojoukko	Kuvaus
<u>tilaus_id</u>	int	Viiteavain Tilaus-tauluun
<u>ongelman_typpi</u>	Edellä mainittu Ongelma_enum, jolla kolme mahdollista arvoa	
ts_ongelma	Timestamp-arvo	Ongelman tapahtuma-ajankohta
ongelman_kuvaus	Merkkijono (hieman pidempi)	Vapaamuotoinen selvitys siitä, mitä tapahtui

Tietokohde: Tuotetyyppi Samasta tuotteesta voi olla useita versioita. Esim. Americano-pizzaa voi olla isoa ja pientä kokoa. Tuotetyyppi on siis tässä tapauksessa Americano-pizza (itse tuote taas on tuotetyypin ja tuoteversion yhdistelmä). Jokaiseen tuotteeseen liittyy yksi tuotetyyppi; jokaiseen tuotetyyppiin liittyy yksi tai useampi tuote. Jokaiseen tuotetyyppiin liittyy tuotekategoria, esim. 'pizza' tai 'virvoitusjuoma'. Tuotekategoria toimii myös viiteavaimena tauluun Hintamuunnos, ts. eri tuotekategorioille pätevät eri hintamuunnokset.

Attribuutti	Arvojoukko	Kuvaus
<u>tuotetyyppi_id</u>	int	Pääavain
tuotekategoria	Tuotekategoria_enum	Viiteavain Hintamuunnokseen. Mahdollisina arvoina 'pizza', 'vegaanipizza', 'virvoitusjuoma', 'olut' ja 'muu'
tuotenimi	Merkkijono	Esim. 'Americano'
tuotekuvaus	Merkkijono	Tuotekuvaus on siis sama kaikille tuotetyypin versioille

4.5.2017

Attribuutti	Arvojoukko	Kuvaus
kuva_tuotteesta	Merkkijono (tiedostopolku, esim. ~/images/kuva.jpg)	

Tietokohde: Tuote

Tuotetta voi ajatella tuotetyypin eräänlaisena ”ilmentymänä”. Tuotteen olemassaolo on riippuvainen tietyistä tuotetyypistä. Tuote yksilöidään sen tuotetyypin ja tuoteversion perusteella.

Attribuutti	Arvojoukko	Kuvaus
<u>tuotetyyppi_id</u>	int	Viiteavain Tuotetyyppi-tauluun
<u>tuoteversio</u>	Tuoteversio_enum	Arvoina 'pieni', 'tavallinen' ja 'iso'
hinta	Numeric(6,2), esim. 123.45	Tuotteen hinta ilman hintamuunnosta

Tietokohde: Tilattu_tuote

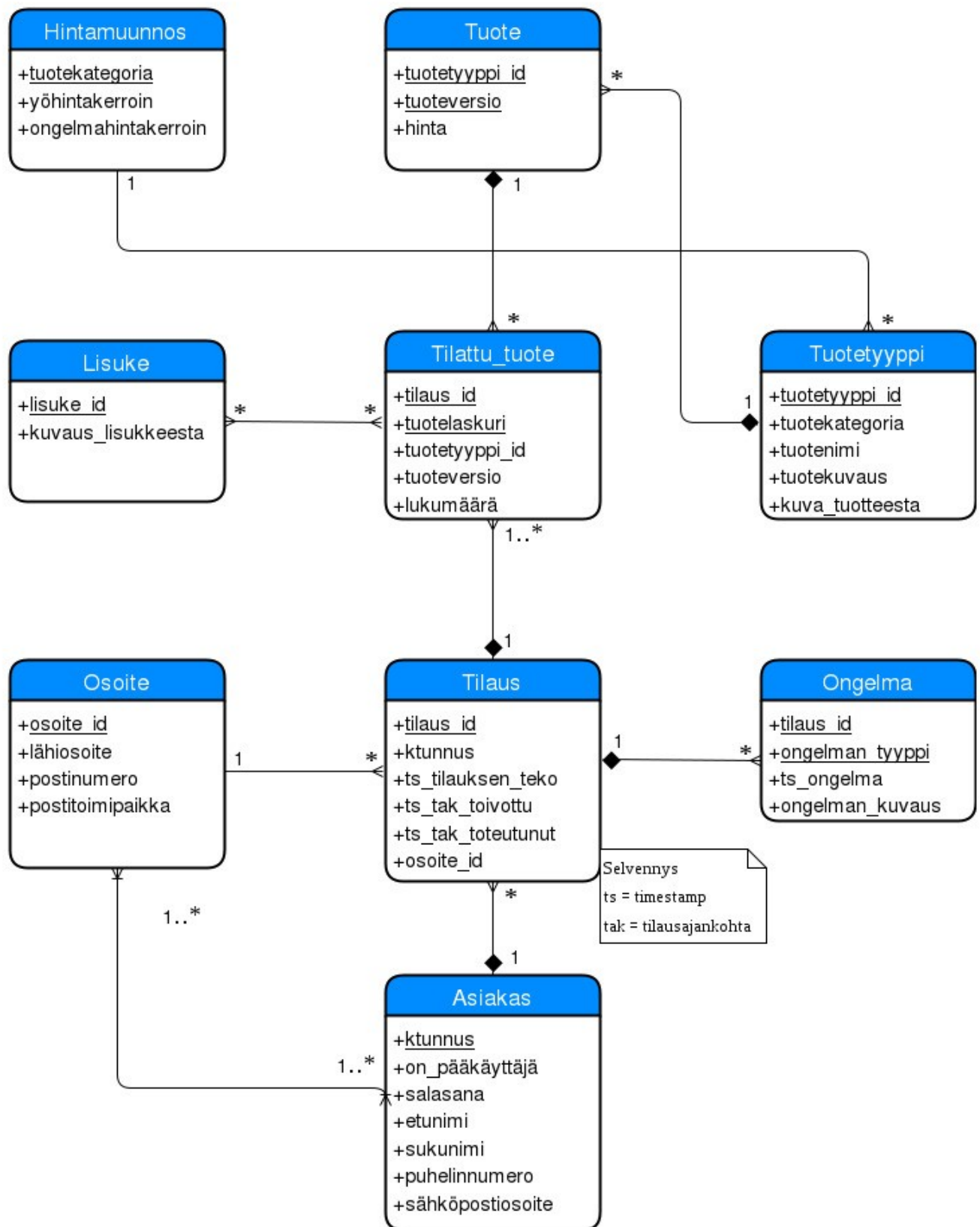
Tilaus koostuu yhdestä tai useammasta tilatusta tuotteesta. Jokainen tilattu tuote liittyy täsmälleen yhteen tuotteeseen.

Attribuutti	Arvojoukko	Kuvaus
<u>tilaus_id</u>	int	Viiteavain Tilaus-tauluun
<u>tuotelaskuri</u>	Serial int	Tilatut tuotteet numeroidaan alkaen ykkösestä
tuotetyyppi_id	int	Yhdessä tuoteversion kanssa viiteavain tauluun Tuote
tuoteversio	Tuoteversio_enum	
lukumaara	int	Esim. 2 * Americano (iso)

Attribuuteilla täydennetty käsitekaavio

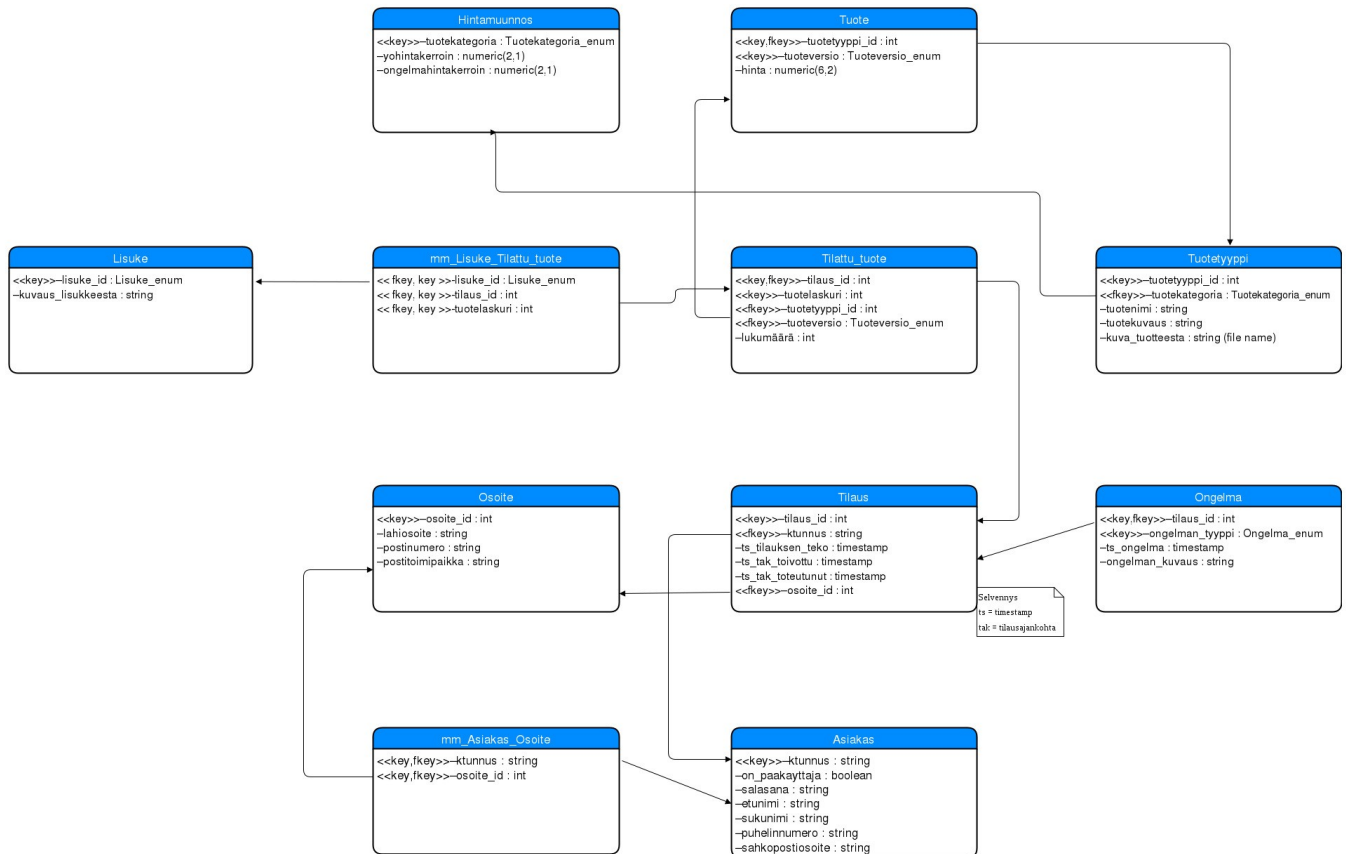
Esitin jo aiemmin Gizzan tietosisällön käsitekaavion avulla. Seuraavalla sivulla on sama käsitekaavio attribuuteilla höystettynä.

4.5.2017



4.5.2017

Relaatiotietokantakaavio Lopuksi vielä relaatiotietokantakaavio, jossa näkyvät myös välitaulut ja viiteavaimet.



Käynnistys- ja käyttöohjeet

(Vanhat käynnistys- ja käyttöohjeet eivät olleet enää ajan tasalla, pitää kirjoittaa uudet lähiaikoina. Olennaisin asia tietää on kai se, että sovellustani voi käyttää joko tavallisena asiakkaana tai ylläpitäjänä. Ylläpitäjän käyttäjätunnus on **admin**. Erään asiakkaan käyttäjätunnus on **mruusu**. Molempien salasana on sama, nimittäin **Tsoh4**.)

Järjestelmän yleisrakenne

Käytin työni pohjana kurssilla tarjottua aloituspakettia [Tsoha-Bootstrap](#). Gizzan hakemistorakenne ja tiedostojen nimet ovat edelleen pitkälti samat kuin aloituspaketissa. Olen mahdollisimman tarkasti pyrkinyt toteuttamaan Gizzan MVC-mallin mukaisesti. Kuten aloituspaketissakin, mallit, näkymät ja kontrollerit löytyvät vastaavasti hakemiston app alikansioista models, views ja controllers. Hakemiston models tiedostot on nimetty niin, että ne vastaavat tietokannan taulujen (eli järjestelmän tietokohteiden) nimiä – esim. taulua Asiakas vastaa tiedosto Asiakas.php. Perusidea on, että kaikki (tai ainakin useimmat) tauluun X liittyvät SQL-kyselyt löytyvät tiedostosta app/models/X.php. Olen välttänyt kokonaan sijoittamasta SQL-kyselyitä muualle kuin hakemiston models tiedostoihin.

Toisin kuin malleissa, kontrollereissa ei ole Gizzassa tarkkaa yksi yhteen -vastaavuutta tietokannan taulujen kanssa. Vaikka olen nimennyt kontrollerit

4.5.2017

taulujen mukaan (esim. asiakas_controller.php), ei kontrollerin toiminta välttämättä rajoitu yksittäiseen tietokohteeseen liittyviin asioihin. Jos Gizzaa haluaisi kehittää ja laajentaa, voisikin olla hyvä miettiä kontrollereille jokin järkevämpi nimeämiskäytäntö kuin se, mitä olen käyttänyt.

Näkymät löytyvät siis hakemistosta app/views. Suurin osa hakemiston HTML-tiedostoista perivät tiedoston base.html. Idea tässä on, että näkymän sivujen yhteinen/yhtenäinen rakenne on määritelty yhdessä paikassa. Olennaisin base.html:n sisältö lienee sivujen ylälaidasta löytyvä navigointivalikko. Hakemistosta views löytyy muutama tiedokohteiden mukaan nimetty alihakemisto. Alikansioiden HTML-tiedostojen nimeämiskäytännöstä on seuraava perusidea:

- index.html – Tietokohteen kaikkien rivien/olioiden listaus
- esittely.html – Yksittäisen olion esittelysivu (olio voidaan usein myös poistaa sivulta löytyvällä painikkeella)
- muokkaa.html – Sivun, jonka avulla tietokohteen oliota voidaan muokata.
- uusi.html – Uuden olion luomista varten (esim. uusi tilaus)

Alihakemistot views/macros ja views/include ovat makroja ja muuta toistuvasti käytettävää HTML-koodia varten. Tärkeitä juurihakemistosta löytyviä hakemistoja ovat app:in lisäksi ainakin config, lib ja sql. Viimeksi mainittu sisältää tarvittavat tiedostot tietokannan taulujen ja niiden alustavan sisällön luomiseen. Seuraavassa joitakin muita tärkeitä tiedostoja:

- config/routes.php – Jokaiseen sovelluksen polkuun (esim. /asiakas/kirjaudu) liittyy tietyn kontrollerin tietty metodi. Ne on määritelty tässä tiedostossa.
- lib/base_controller.php ja lib/base_model.php – Kaikkien Gizzan kontrollerien ja mallien yläluokat. Kaikille kontrollereille tai malleille yhteisiksi tarkoitettut funktiot ja attribuutit sijoitetaan näihin tiedostoihin.

Katsotaan vielä Gizzan keskeisiä istuntoihin liittyviä funktioita:

- asiakas_controller.php: sisaankirjautumisen_kasittely() – Jos todennus onnistuu, asetetaan superglobaalin muuttujan \$_SESSION['user'] arvoksi asiakkaan käyttäjätunnus
- Asiakas.php: todenna(\$ktunnus, \$salasana) – Jos parametrit täsmäävät, palauttaa vastaavan Asiakas-olion. Annetun käyttäjätunnuksen ja salasanan oikeellisuuden tarkistus tapahtuu siis tämän funktion avulla.
- base_controller.php – Kaikki tämän luokan funktiot ovat olennaisia istuntojen tai käyttöoikeuksien tarkistamisen kannalta

Käyttöliittymä ja järjestelmän komponentit