

Pizzapalvelu

Marko Hassinen
014029585

Tietokantasovellus
Syksy 2013

Sisällysluettelo

Johdanto	3
Yleiskuvaus järjestelmästä	4
Käyttötapaukset	5
Järjestelmän tietosisältö	6
Tietokohteet	7
Relaatiotietokantakaavio	10
Järjestelmän yleisrakenne	11
Järjestelmän komponentit	11
Sivukartta	18
Asennustiedot	19
Käyttöohje	19
Testaus, tunnetut bugit ja jatkokehitysideat	20
Omat kokemukset	21
Liitteet	21

Johdanto

Työn aiheeksi on valittu kurssimateriaalin Pizzapalvelu-tehtävä, jonka tarkoituksena on luoda virtuaalinen tilausjärjestelmä. Järjestelmässä tarjottavat tuotteet, kuten pizzat ja juomat, on jaettu eri tuoteryhmiksi. Kutakin tuotetta esitellään tuotekuvauksessa, osasta tuotteita on olemassa myös kuva. Järjestelmän mukaisesti valmiiksi suunniteltuihin pizzoihin voi halutessaan valita lisukkeita.

Pizzapalvelussa ilmaistaan tuotteiden hinnat, joiden muuttumista koskeva oikeus on säilytetty. Laskutuksessa huomioidaan kuitenkin tilausajankohtainen hinnasto.

Toimitusaika on ennalta sovittu. Jos toimitus sovittuun aikaan ei toteudu, asiakkaalle voidaan myöntää alennusta tilauksesta. Toimituksessa tapahtuneet häiriöt voivat johtaa tilauksen eväykseen. Asiakkaan on mahdollista purkaa tilauksensa, kuitenkin viimeistään tunti ennen sovittua toimitusaikaa.

Alla oleva toimintolista on lainattu osoitteesta

<http://advancedkittenry.github.io/aiheet/Pizzapalvelu.html> (luettu 3.11.2013, 15:07).

Toimintoja:

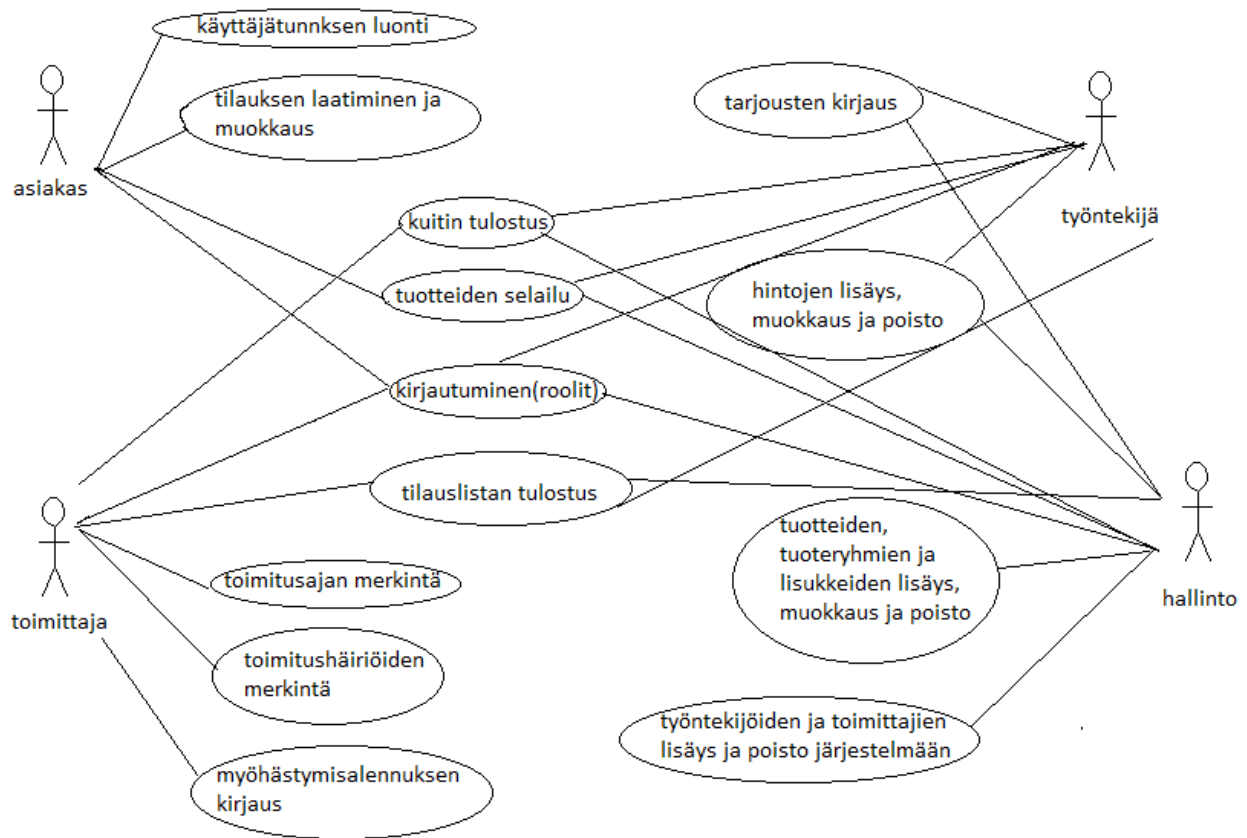
- tuotetietojen ja lisukkeiden lisäys, muokkaus ja poisto
- ajasta riippuvien hintojen lisäys, poisto ja muokkaus
- tuotteiden selailu
- tilauksen laatiminen ja muokkaus
- toimituksen ja mahdollisen myöhästymisalennuksen kirjaus
- tilauslistan tulostus
- kuitin tulostus
- tarjousten kirjaus
- ylläpidon kirjautuminen ja tilauksen tehneen asiakkaan tunnistaminen/kirjautuminen

Työssä hyödynnetään versionhallintaa, ja työ on nähtävissä osoitteessa

<https://github.com/mihassin/tietokantasovellus-s2013>. Toimiva ohjelmisto on tarkoitus laittaa users-palvelimelle. Työ toteutetaan PHP-ohjelmointikielellä. Työssä hyödynnetään tietokannan hallintajärjestelmää PostgreSQL. Alustavasti työ mahdollistaa vain yhden tietokannan käytön.

Yleiskuvaus järjestelmästä

Käyttötapauskaavio:



Käyttötapaukset

Asiakas

Asiakas avaa pizzapalvelun selaimellaan. Jos asiakkaalla ei ole käyttäjätunnuksia palveluun, voi hän luoda itselleen sellaisen. Järjestelmään kirjautumisen jälkeen asiakas voi selata tuotelistaa ja valita haluamansa tuotteet mahdollisine lisukkeineen tilaukseensa. Asiakastiedot saadaan käyttäjätiedoista. Asiakas merkitsee tilaukseen järjestelmän hyväksymän toimitusajan ja toimitusosoitteen. Tilausta voidaan muokata tai se voidaan poistaa vähintään tunti ennen sovittua toimitusaikaa.

Toimittaja

Toimittaja kirjautuu järjestelmään toimittajana ja tulostaa tilauslistan ennen toimitusta. Tilaus toimitetaan tilauksessa mainittuun toimitusosoitteeseen ja toimitus pyritään toteuttamaan sovittuun aikaan. Toimittaja merkitsee toimitusajan ja mahdolliset toimitushäiriöt sekä myöhästymisalennuksen. Toimittaja tulostaa asiakkaalle kuitin, johon on merkitty mahdollinen alennus.

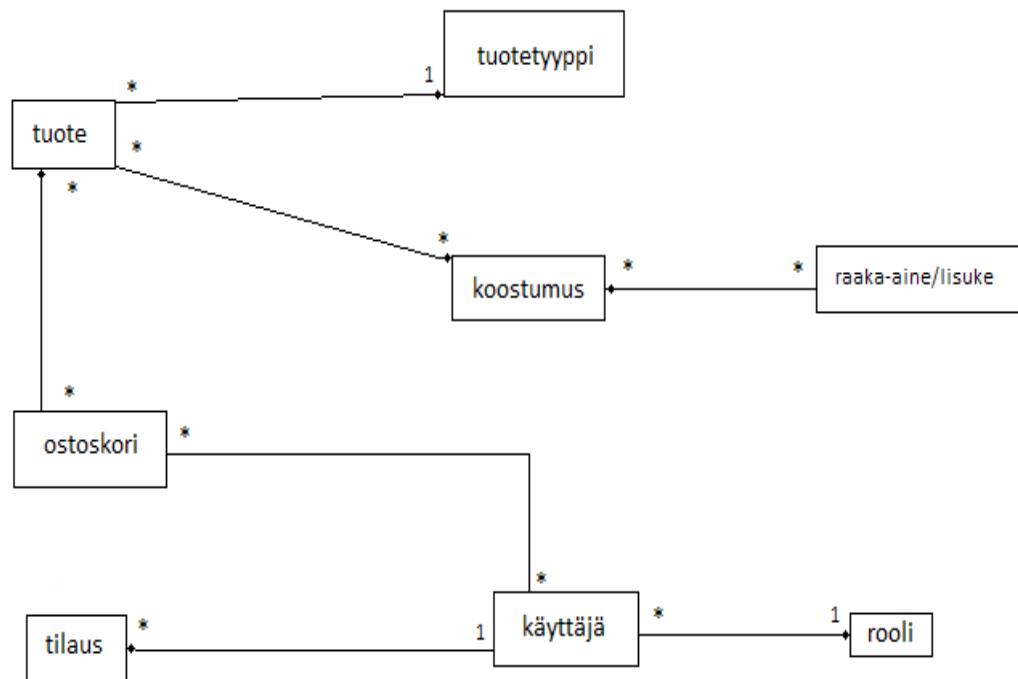
Työntekijä

Työntekijä kirjautuu järjestelmään työntekijänä. Työntekijän on kyettävä päivittämään hinnastoa ja merkitsemään tarjoukset järjestelmään. Lisäksi työntekijän pitää pystyä monitoroimaan sekä tuote- että tilauslistaa ja tarvittaessa tulostamaan kuitteja tilauksista.

Hallinto

Hallintoroolissa ei ole merkittäviä eroja työntekijän rooliin, mutta hallinnon on pystyttävä lisäämään, muokkaamaan ja poistamaan tuotteita, tuoteryhmiä tai lisukkeita. Hallinto myös sekä lisää että poistaa työntekijöitä ja toimittajia järjestelmään.

Järjestelmän tietosisältö



Tietokohteet

Työssä on kahdeksan tietokohdetta, joista kaksi on “mappereita”. Tietokohteille 'käyttäjä' ja 'tuote' on käytössä pienempiä hallinnointitaulukoita. Muutamalle tietokohteelle on luotu taulukko, josta ilmenevät avaimia lukuun ottamatta muut attribuutit.

Käyttäjä

Käyttäjä		
Attribuutti	Arvojoukko	Kuvailu
Etunimi	30 merkkiä pitkä merkkijono	Ilmaisee käyttäjän etunimen
Sukunimi	30 merkkiä pitkä merkkijono	Ilmaisee käyttäjän sukunimen
Sähköposti	255 merkkiä pitkä merkkijono	Ilmaisee käyttäjän sähköpostiosoitteen
Puhelin	255 merkkiä pitkä merkkijono	Ilmaisee käyttäjän puhelin numeron

Käyttäjille on oma yksilöivä id-attribuutti ja rooli-id-attribuutti, jolla ilmaistaan tietokohteiden 'käyttäjä' ja 'rooli' välinen relaatio. Sähköposti toimii järjestelmässä käyttäjätunnuksena. Salasana salataan tietokantaan, ja salauksen purkua varten on oma suola.

Rooli

Tietokohde 'rooli' toimii 'käyttäjä'-tietokohteen apuna yksilöimässä käyttäjien roolia järjestelmässä. Tietokohteella on kaksi attribuuttia: avain ja pienimuotoinen kuvaus.

Tuote

Tuote		
Attribuutti	Arvojoukko	Kuvailu
Nimi	30 merkkiä pitkä merkkijono	Ilmaisee tuotteen nimen
Kuvaus	255 merkkiä pitkä merkkijono	Mahdollista lisätietoa tuotteesta
Hinta	Reaaliluku	Ilmaisee tuotteelle merkityn hinnan

Tietokohde ilmaisee tuotteita, joita pizzapalvelu tarjoaa asiakkailleen. Palvelu vastaa tuotteen valmistuksesta ja toimituksesta. Tietokohteella on attribuuttina yksilöivä avain ja tuotetyypin avain, joka ilmaisee tietokohteiden välisen relaation.

Tuotetyyppi

Kyseessä on pienempi tietokohde, joka auttaa palvelun tuotteiden ja raaka-aineiden järjestelyssä. Tuotetyypillä on kaksi attribuuttia: yksilöivä avain ja pienimuotoinen kuvas.

Raaka-aine

Raaka-aine		
Attribuutti	Arvojoukko	Kuvailu
Kuvaus	30 merkkiä pitkä merkkijono	Kertoo raaka-aineen nimen
Hinta	Reaaliluku	Kertoo raaka-aineen hinnan

Tietokohdetta 'raaka-aine' käytetään tuotteiden koostumuksen ilmaisemiseen. Järjestelmä sallii raaka-aineiden lisäyksen tuotteisiin. Tietokohteella on attribuutteina yksilöivä avain ja tuotetyypin avain.

Koostumus

'Koostumus'-tietokohdetta käytetään pizzatuotteiden koostumuksen ilmaisuun. Tietokohde toimii niin sanottuna mapperina, jolla liitetään raaka-aineet tuotteeseen. Tietokohteella on kolme attribuuttia: oma avain, tuotteen avain ja raaka-aineen avain.

Ostoskori

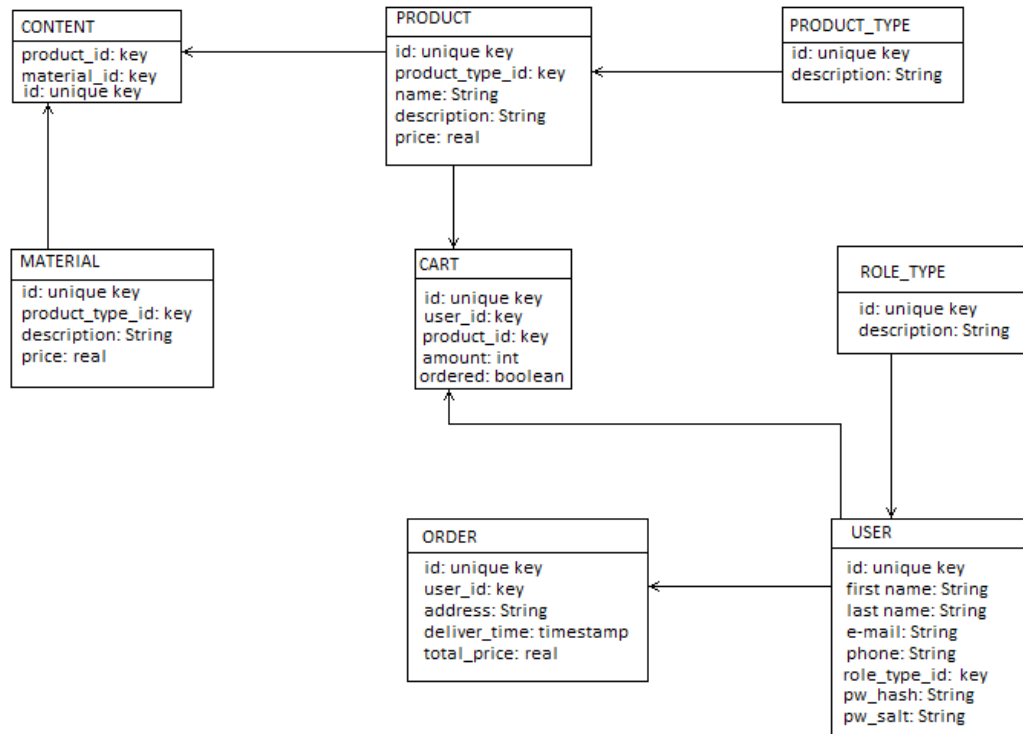
Ostoskori on järjestelmän toinen “mapperi”. Sillä ilmaistaan tilauksen sisältö. Tietokohteella on neljä attribuuttia: oma avain, käyttäjän avain, tuotteen avain, määrä (yhden tuotteen määrä) ja totuusarvo ordered.

Tilaus

Tilaus		
Attribuutti	Arvojoukko	Kuvailu
Toimitusosoite	255 merkkiä pitkä merkkijono	Kertoo asiakkaan valitseman toimitusosoitteen
Toimitusaika	Päivämäärä ja kellonaika	Kertoo asiakkaan valitseman toimitusajan
Yhteishinta	Reaaliluku	Toimituksen lopullinen hinta

'Tilaus'-tietokohde on tarkoitettu käyttäjille, jotka eivät ole asiakkaita. Tietokohteella on attribuuttina oma yksilöivä avain ja käyttäjän avain.

Relaatiotietokantakaavio



Järjestelmän yleisrakenne

Järjestelmän juurikansio, htdocs, sisältää kansiot sql, libs, css ja views sekä osan php-tiedostoista. Kansio sql koostuu kaikista tietokantojen alustukseen ja poistoon tarvittavista tiedostoista. Tyylitiedostot löytyvät css-kansiosta. Libs-kansiossa on php-tiedostoja, jotka tarjoavat järjestelmässä yleisesti käytettyjä toimintoja ja tietokannan hallintatoimintoja. Views-kansion php-tiedostot puolestaan ovat pääosin vastuussa ”html-tägeistä”, joilla määritetään sivujen rakenne. Htdocs-kansion php-tiedostot kokoavat tarvittavat views-kansion tiedostot, liittävät libs-kansion toiminnallisuuden sivuihin ja määrittävät kyselyt, joita käytetään libs-kansion funktioissa.

Järjestelmän komponentit

index.php

Sivuston etusivu, joka käsittää näkymät (views/) header.php, etusivu.php ja footer.php.

etusivu.php

Etusivun html-rakenteesta vastaava tiedosto, joka sijaitsee yleiskäyttöisten näkymätiedostojen header.php ja footer.php välissä, views-kansiossa. Tiedosto sisältää linkit ruoka-, juoma- ja lisukelistaan.

views/header.php

Toteuttaa sivuston navigaatioelementin, joka näkyy jokaisella sivulla. Elementin sisällössä voi olla sisäänkirjautumattoman käyttäjän tai sisäänkirjautuneen käyttäjän roolin määrittämiä eroja. Jos käyttäjä on sisäänkirjautunut, header.php-tiedosto ohjaa tiedostoon, josta navigaatioelementti haetaan. Jos käyttäjä ei ole kirjautunut sisään, header.php vastaa navigaatioelementin sisällöstä.

customer-header.php/admin-header.php/employee-header.php/delivery-header.php

Sivustolla käytetään neljää eri roolia, ja edellä mainitut komponentit vastaavat kunkin roolin navigaatioelementin sisällöstä. Komponentit sijaitsevat views-kansiossa.

views/footer.php

Vastaa jokaisen sivun alareunaan sijoittuneen marginaalin sisällöstä. Sivuston on suunniteltu näyttävän tässä elementissä sivuston päivityksestä kertovaa informaatiota, mutta marginaaliin voidaan liittää myös esimerkiksi yhteystietoja tai muuta käyttäjälle hyödyllistä informaatiota.

menu.php/drinks.php/materials.php

Sijaitsevat juurikansiossa ja ovat vastuussa näkymien sekä funktioiden kasaamisesta sivulle. Jokainen tiedosto vie uudelle sivulle sivustolla. Tiedostot myös kutsuvat libs/listing.php-tiedoston funktiota getList(), jonka tehtävä on listata tietokannasta tarvittava tieto sivulle.

Tiedostoihin sisältyy kyselymerkkijonoja, joista osa haetaan funktioista libs/listqueries.php, ja ne mahdollistavat näkymän listatietojen lisäys-, poisto- ja muunnostoiminnoille. Toiminnot kuitenkin näkyvät vain, mikäli käyttäjän rooli sen sallii.

Views-kansiossa on jokaista edellä mainittua tiedostoa vastaava samanniminen tiedosto, joissa on rivi html-merkkauksia. Tämä johtuu siitä, että järjestelmässä on tarkoitus erottaa mahdollisimman hyvin html-merkkaukset views-kansioon.

libs/listqueries.php

Tiedosto sisältää kysely merkkijonoja.

libs/listing.php

Tiedostossa on funktio getList(\$kysely), joka listaa parametrin \$kysely mukaisen tietokantakyselyn tietoja html:n taulukkoesityksenä. Tiedosto käyttää libs/db_connect.php-yhteyden muodostamiseen tietokantaan.

libs/db_connect.php

Tiedosto sisältää funktion getConnection(), jonka on tarkoitus palauttaa yhteys kysyvälle tiedostolle. Tiedosto saa tarvittavat tiedot tietokanta yhteyden muodostamiseen tiedostolta secret.php, joka sijaitsee libs kansiossa.

views/listbuttons.php

Tiedosto sisältää ”nappiesityksen”. Nappeja on kolme – lisäys-, poisto- ja muunnosnapit – ja ne vievät uudelle sivulle. Lisäys- ja poistonapit näkyvät vain ’admin’-tason roolille, muunnosnapit työntekijöille ja kuljetukselle. Poistonapin lomake sisältää listausesityksen, jossa valitaan poistettava tuote listasta. Listaus saadaan tiedoston libs/dropList.php funktiosta getDropList(). Lisäysnappi vie juuren tiedostoon add-item.php, muutosnappi tiedostoon add-offer.php.

libs/dropList.php

Otaa yhteyden tietokantaan db_connect.php-tiedoston avulla ja listaa kyselyn mukaiset optiot listaan.

login.php

Lisää tiedoston views/login.php sisältämän lomakkeen.

views/login.php

Sisältää html kaavakkeet kirjautumista varten. Kaavakkeen lähetys vie juurikansion tiedostoon checklogin.php

checklogin.php

Tiedosto muodostaa yhteyden tietokantaan libs/db_connect.php-tiedoston kautta. Kirjautumistietoina käytetään käyttäjän salasanaa ja sähköpostiosoitetta. Tiedot tarkistetaan tietokannasta. Mikäli tiedot ovat virheellisiä, käyttäjälle näytetään views/failure.php-tiedoston sisältö. Jos tiedot ovat oikeita, aloitetaan kirjautumissessio, jossa pidetään kirjaa käyttäjän roolista ja tunnuksesta. Lopuksi käyttäjä ohjataan sivuston etusivulle.

views/failure.php

Tiedosto ilmoittaa kirjautumistietojen virheellisyydestä, muttei kerro käyttäjälle erityisesti missä oli vikaa tietoturvalisistä syistä. Käyttäjä näkee myös linkin jota kautta on mahdollista palata etusivulle.

logout.php

Päättää sisäänkirjautumisesta syntyneen session.

users.php

Tarjoaa taulukoituja tietoja käyttäjistä. Tietojen listaukseen käytetään libs-kansion listing.php-tiedostoa, jonka käyttämä kysely saadaan libs-kansion listqueries.php-tiedostosta. Sivun otsikko saadaan samannimisestä tiedostosta users.php, joka on views-kansiossa.

orders.php

Tarjoaa taulukoituja tietoja tilauksista. Tietojen listaukseen käytetään libs-kansion listing.php-tiedostoa, jonka käyttämä kysely saadaan libs-kansion listqueries.php-tiedostosta.

newuser.php

Sivun sisältö haetaan views-kansion samannimisestä tiedotosta newuser.php.

views/newuser.php

Tiedosto sisältää html-lomakkeen uuden käyttäjätunnuksen luontia varten. Mikäli jo sisäänkirjautunut käyttäjä yrittää päästä sivulle, hänet ohjataan takaisin etusivulle. Lomakkeen tiedot lähetetään juurikansion tiedoston newuserCheck.php tutkittavaksi.

newuserCheck.php

Sisäänkirjautunut käyttäjä ohjataan etusivulle. Tiedosto ottaa yhteyden tietokantaan libs-kansion db_connect.php-tiedoston kautta ja tutkii uuden käyttäjän lomakkeen tiedot. Mikäli lomakkeessa annetut tiedot ovat virheellisiä, siitä kerrotaan ja näytetään views-kansion newuserAgain.php-tiedoston sisältämä linkki tunnuksen luomista varten.

Sivusto sallii kullekin sähköpostiosoitteelle vain yhden käyttäjätunnuksen. Varattu sähköpostiosoite tutkitaan libs-kansion checkEmail.php-tiedoston avulla. Jos tiedot ovat kunnossa, luodaan kysely, joka lähetetään tietokantaan. Lopuksi käyttäjälle ilmoitetaan tunnuksen luonnin onnistumisesta ja tarjotaan linkki etusivulle.

views/newuserAgain.php

Sisältää linkin juuren newuser.php-tiedostoon.

remove-itemCheck.php

Tiedosto vastaa tuote-, juoma- ja lisuke listojen poisto toiminnon kyselyiden suorittamisesta tietokantaan. Sivulla ei ole esitystä ja käyttäjä ohjataan välittömästi etusivulle.

add-offer.php

Hakee tiedostot checkAccess.php ja dropList.php kansista libs. Sivun sisältö on views kansion add-offer.php tiedostossa.

views/add-offer.php

Sisältää html-lomakkeen tuotteiden hintamuutoksia varten. Lomakkeen tarkistus ohjataan juuren tiedostoon add-offerCheck.php

add-offerCheck.php

Tiedosto luo yhteyden tietokantaan libs-kansion db_connect.php-tiedoston avulla. Libs-kansion tiedosto checkAccess.php ohjaa sisäänkirjautumattomat ja 'asiakkaat' etusivulle. Mikäli hinta ei ole numeerinen, ohjataan käyttäjä takaisin lomakkeeseen. Lomakkeen tietojen pohjalta päivitetään tietokantaan tiedot. Lopuksi käyttäjä ohjataan etusivulle.

add-item.php

Hakee tiedostot checkAccess.php ja dropList.php kansiosta libs. Sivun sisältö on views-kansion add-item.php-tiedostossa. Sisältöä varten luodaan kyselymerkkijono.

views/add-item.php

Sisältää html-lomakkeen uuden tuotteen luomista varten. Lomake mahdollistaa jokaisen tuotetyypin luomisen. Lomakkeen tarkistus ohjataan juuren tiedostoon add-itemCheck.php.

add-itemCheck.php

Tiedosto luo yhteyden tietokantaan libs-kansion db_connect.php-tiedoston avulla. Libs-kansion checkAccess.php ohjaa sisäänkirjaantumattomat ja asiakkaat etusivulle. Mikäli hinta ei ole numeerinen ohjataan käyttäjä takaisin lomakkeeseen. Lomakkeen tietojen pohjalta päivitetään tietokantaan tiedot. Tietokanta kysely eroaa lisukkeiden ja muiden tuotetyyppien välillä. Lopuksi käyttäjä ohjataan etusivulle.

libs/checkAccess.php

Ohjaa sisäänkirjautumattomat ja 'asiakaat' etusivulle

libs/checkEmail.php

Sisältää funktion emailExists(\$email), joka tarkistaa, kuinka monta parametrin \$email mukaista sähköpostiosoitetta on tietokannassa. Yhteys tietokantaan luodaan db_connect.php-tiedoston kautta.

cart.php

Huolehtii ostoskorin esityksen organisoinnista. Näkyvyys vain kirjautuneille. Listauksen suorittaa libs-kansion tiedosto list-cart.php. Kyselyt noudetaan libs-kansion listqueries.php-tiedostosta. Poisto- ja lisäysnapit haetaan views-kansion cartbuttons.php-tiedostosta.

libs/list-cart.php

Sisältää funktion getCartList(\$kysely), joka palauttaa parametrin \$kysely mukaisen esityksen ostoskorin listaukselle. Jos ostoskori on tyhjä, siitä ilmoitetaan.

views/cartbuttons.php

Tulostaa ostoskorisivulle toiminnallisuutta tarjoavat napit “Lisää tuote”, “Lisää juoma”, “Poista tuote” ja “Tilaa”. Näistä kaksi jälkimmäistä näkyy käyttäjälle vain, jos ostoskorissa on vähintään yksi tuote.

add-item-cartCheck.php

Tiedosto tarkistaa ostoskorin “Lisää tuote”-napin lomakkeen ja päivittää tietokannan. Päivityksen jälkeen käyttäjä ohjataan takaisin ostoskorisivulle.

add-drink-cartCheck.php

Tiedosto tarkistaa ostoskorin “Lisää juoma”-napin lomakkeen ja päivittää tietokannan. Päivityksen jälkeen käyttäjä ohjataan takaisin ostoskorisivulle.

remove-item-cartCheck.php

Tiedosto tarkistaa ostoskorin “Poista tuote”-napin lomakkeen ja päivittää tietokannan. Päivityksen jälkeen käyttäjä ohjataan takaisin ostoskorisivulle.

removeCart.php

Tiedosto tyhjentää kirjautuneen käyttäjän ostoskorista tilaamattomat tuotteet.

libs/cart-total.php

Tiedosto laskee kirjautuneen käyttäjän valitsemien tuotteiden yhteishinnan. Tietoa käytetään tilauksen arvon laskemiseen.

libs/check-data.php

Tiedosto sisältää funktion checkData(\$data), joka suorittaa kolme funktiota annetulle parametrille \$data. Funktiot ovat trim(), stripslashes() ja htmlspecialchars(). Tiedostoa hyödynnetään käyttäjien syötteiden tarkistukseen, jolla pyritään estämään XSS-hyökkäykset.

addMaterialsToItems.php

Tiedosto kokoaa tarvittavat funktiot libs kansion tiedostoista ja näkymän addMaterialsToItems.php views kansista.

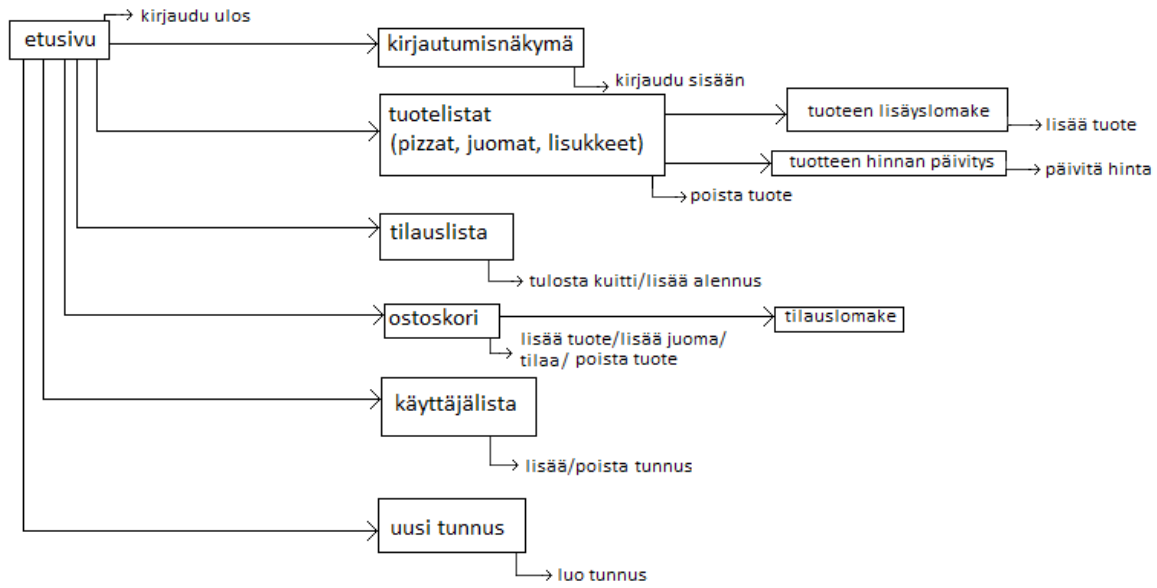
views/addMaterialsToItems.php

Tiedosto merkitsee näkymän lomaketta varten. Lomakkeella lisätään lisukkeita tuotteisiin. Lisukkeita voi korkeintaan olla neljä kappaletta tuotetta kohden. Lomakkeen syötteet ohjataan juurikansion check-mats.php tarkastettavaksi.

check-mats.php

Tiedosto tarkistaa views-kansion näkymän addMaterialsToItems.php lomakkeen syötteet. Syötteiden pohjalta luodaan sql-kysely tietokantaan, jolla päivitetään tietokannan tietokohdetta cart_map.

Sivukartta



Asennustiedot

Sivuston käyttö edellyttää kaikkien siihen sisältyvien tiedostojen lisäämistä, mikä tulisi ottaa huomioon, jos projektia hyödynnetään harjoitustyön ulkopuolisessa yhteydessä.

Jos sivusto halutaan siirtää, tulee siinä olevia tietoja muuttaa niin, että yhteyden muodostaminen tietokantaan on mahdollista. Asennuksessa suositellaan harjoitustyössä käytettyä kansiorakennetta. Mikäli kansiorakenteeseen halutaan tehdä muutoksia, tulee huomioida mahdolliset muutokset tiedostojen osoittamisessa ja tiedostoissa, jotka osoittavat muihin tiedostoihin.

Oletuksena on, että tietokannan hallintajärjestelmä PostgreSQL ja ohjelmointikieli PHP ovat käytettävissä palvelimella, jolle sivusto asennetaan. Sivustoa käytetään oletuksellisesti selaimella, mutta selainohjelmaan ei oteta kantaa.

Vaikka sivusto on suunniteltu pizzeriayrityksen sähköiseen palveluun, voidaan järjestelmän käyttö yleistää pienin muutoksin muillekin ravintolayrityksille. Yksinkertaisten muutosten avulla on myös mahdollista kohentaa sivuston ulkoasua. Muutokset tehdään views/header.php-tiedostoon, jonne merkitään sivustossa hyödynnettävät tyylitiedostot. Tyylitiedostot tulisi asentaa css/-kansion alle.

Käyttöohje

Harjoitustyö löytyy osoitteesta <http://mihassin.users.cs.helsinki.fi/>. Sivuston käyttäminen ei edellytä kirjautumista, mutta kirjautuminen takaa laajemman joukon sivuston käytettömahdollisuuksia.

Kirjautumisessa huomioidaan käyttäjän rooli, joka määrää käyttäjälle tarpeelliset käyttösovellukset. Sivustolla on käytössä neljä eri roolia, joista jokaiselle tarjotaan testausmahdollisuutta. Alla olevassa taulukossa esitellään salasanoineen sivuston testaukseen suunniteltuja tunnuksia, jotka eroavat rooleiltaan.

tunnus	salasana	rooli
admin@test	test	admin
delivery@test	test	kuljetus
employee@test	test	työntekijä
user@test	test	käyttäjä

Testaus, tunnetut bugit ja jatkokehitysideat

Sivusto ja sen toiminnallisuus on käsintestattu. Mitään testaus ohjelmistoa ei ole käytetty taikka luotu. Suurin osa testaamisesta on kohdistunut lomakkeiden täyttämiseen. Ostoskorin listausta ja käyttäytymistä on testattu eniten useilla eri syötteillä ja käyttötilanteilla.

Ostoskorin yksittäisen tuotteen poistossa on ongelmia. Poistettava tuote valitaan listasta, joka merkitään '<select>'-ilmauksella. Listattavat elementit merkitään '<option>'-ilmauksella, jonka attribuuttina 'value' ilmaistaan valittavan elementin 'value'-arvoa. Tällä hetkellä 'value' ilmaisee ostoskorin tuotteen nimeä. Tilanne, jossa käyttäjä on valinnut ostoskoriinsa kaksi samannimistä tuotetta, aiheuttaa poistossa virheellistä toimintaa. Listauksessa ei ole virhettä, vaan sen arvon ilmaisussa. Jos poistetaan tuote ja ostoskorissa on samanniminen tuote, molemmat tuotteet katoavat ostoskorista, vaikka vain yhden piti poistua. Listauksessa tulisi jotenkin indeksoida tai erotella tuotteet, esimerkiksi id-kentän avulla.

Virheilmoitusten järkevä ja käyttäjäystävällinen toteutus lomakkeiden täytössä osottautui kaikista haastavammaksi osaksi työtä. Tätä tulisi luonnollisesti parantaa. Sivusto on visuaalisesti tosi vaisu ja kaipaasi tyylikäämpää toteutusta. Tähän ei keskitytty työn suorituksessa, koska koettiin toiminnallisuuden olevan tärkeämpi tavoite. Toiminnallisuuden toteutus söikin suurimman osan ajasta. Sivustoa ei ole sidottu valuuttaan ja mikäli sivustoa haluttaisiin käyttää kaupallisessa yhteydessä, tulisi tämä korjata. Sivuston ohjelmoinnin rakenne ei noudata mvc-mallia kokonaisuudessaan ja on osittain epäjohdonmukainen. Suurin syy tähän on kokoajan tottumattomuus työssä käytettyihin työkaluihin ja niissä hyödynnettäviin menetelmiin.

Tilauksen seuranta olisi todella hyödyllinen ominaisuus. Sivusto voisi myös ilmoittaa tilaajille käyttäjätunnuksissa annettuihin sähköpostiosotteisiin tilauksen vastaanottamisesta ja toimituksen aloituksesta. Ruokalistan tuoteryhmiä voisi lisätä.

Omat kokemukset

Suurin haaste työssä oli php-ohjelmointikielten syntaxin ja semantiikan opettelu lyhyessä ajassa. Osa aikataulun tehtävistä jäi tämän syyn takia suorittamatta. Java on tullut tutuksi jo usean kurssin myötä, mutta php:ta en ole oppinut aiemmilla kursseilla. Ymmärsin lähteitteni pohjalta, että php-ohjelmoinnissa on tyylillesti kaksi koulukuntaa, OOP ja imperatiivinen ohjelmointi. Lähestyin jälkimmäisellä tavalla, koska aikaisemmassa periodissa osallistuin c-ohjelmointi kurssiin. Huomasin valitettavasti myöhemmin, että tätä oli vaikea implementoida kurssilla pyydettyyn mvc-malliin ja päätin suuremmalta osin jättää tämän huomiotta. Käytin kurssin aikana ensimmäistä kertaa tietokannanhallintajärjestelmää ja tähän totuttelu vei aikansa.

Koin 'demo-tilaisuuden' todella arvokkaksi, kun sain ulkopuolista kritiikkiä sivustosta vertaisiltani. Viikottainen palaute oli joskus niukkaa ja tilaisuus avasi silmäni. Toisten töiden näkeminen myös motivoi parantamaan omaa työtäni. Päätin kurssin aikana olla osallistumatta 'paja-työskentelyyn', sillä koen suoriutuvani yksin paremmin. Myöhemmin kuitenkin ajattelen, että ehkä kerran viikossa järjestettävä 'demo-tilaisuuden' kaltainen tilaisuus olisi auttanut työn teossa.

Ohjelmoinnissa en kokenut ongelmia ja ehkä se oli helpointa. Koska testasin käsin sivustoa, niin virheiden ja sivuston virheellisen toiminnan löytäminen oli vaikeaa. Juuri tässä asiassa koin 'demo-tilaisuuden' arvon. Esimerkiksi en tullut ajatelleeksi, että käyttäjä voi halutessaan tilata negatiivisen arvon tuotteita ja sivusto salli tämän.

Liitteet

Työn tiedostot: <https://github.com/mihassin/tietokantasovellus-s2013>

Create table lauseet: <https://github.com/mihassin/tietokantasovellus-s2013/blob/master/sql/create-tables.sql>

Tehtäväkuvaus: <http://advancedkittenry.github.io/aiheet/Pizzapalvelu.html> (luettu 3.11.2013, 15:07)

Sivusto: <http://mihassin.users.cs.helsinki.fi>