Systemdokumentation

API

Systemet använder sig av ett API för att kommunicera med databasen och hantera data.

Källkoden till API:t finns i mappen ”matkasseWS” i projektets huvudmapp. Koden exporteras dock ut i en .war fil som installeras på tomcat servern och kör API:t på det sättet.

Vårt API är byggt på Jersey RESTful WebService som är ett tredjeparts bibliotek som gör det lätt att bygga en webservice. Man bara importerar dessa bibliotek i sitt projekt och sedan skapar man Java filer som hanterar metoder beroende på URL:en som skickas in.

Vi har byggt det så att varje databasobjekt har en egen Java-klass som säger vilka fält objektet behöver. Sedan har varje objekt en s.k. "resource" fil som innehåller URL:erna som används och metoder som tillhör. Dessa resource filer kastar tillbaka information i form av antingen en string (om vi gör en update, add eller delete) eller JSON-objekt om vi vill hämta data.

För att sedan använda API:t i webklienten så behöver man bara gå till URL:erna som finns i "resource" filerna och då kallas metoderna som tillhör och skickar tillbaka informationen som hör till.

Webb-Klienten

Källkoden för webbklienten finns i mappen public och lite i mappen innan.  
Mappen innan public är mappen innehåller header, config och footer. D.v.s filerna som inkluderas i alla dokument och därför finns de utanför resten av koden. Dessa filer innehåller information som ska finnas med i varje sida som headern och footern på sidan. Config filen innehåller konfigurationsinformation för sidan.

I samma mapp finns classes mappen som innehåller databaskod och loginkod osv. Denna kod används inte så mycket utan mer bara vid specifika fall. För det mesta används API:t.

Huvudstrukturen är att man börjar med att komma till index.php där man kan logga in. Efter man har loggat in så ser vi vilket privilege användaren har och sedan skickar användaren till motsvarande sida. Antingen administrator.php, supplier.php eller index.php för att ge ett par exempel.

Efter att man har loggat in så fungerar det som så att när vi ska kalla på någon API funktion som gör en delete, update eller add så kallas det en run.php fil som innehåller alla kallelser till API:t genom att ange vilken funktion man vill ha och ger sedan en respons tillbaka.  
Om man som admin vill ta bort en category så skickar man användaren till ”run.php?func=adm\_delete\_cat” så körs den funktionen i run.php filen och gör det den ska och skickar sedan användaren tillbaka dit den var.

Så vi har filer/sidor för varje moment som till exempel ”new\_product.php” där man som grossist kan lägga till ny vara och när man trycker på ”add” knappen så körs run.php med tillhörande funktion

Android

Källkoden till android-appen finns i mappen ”RoyalHearts” i projektets huvudmapp.  
Appen använder sig också av API:t för att hämta information från databasen.

Appen börjar med att kalla på login.java där man måste logga in. Efter det att man loggat in skickas man till MainActivity.java som är huvudaktiviteten för appen. Där man kan välja stad och butik och få fram sin inköpslista för vald butik. Man kan med actionbaren gå till en ny aktivitet för att lägga till produkter. Man kan även gå in på inställningar med actionbaren och där kan man se sina användaruppgifter, logga ut och läsa om appen.