Bakery Recipe Management System

Dieter Zancanella

Tutor: Michael Wild

Class 5 I a

TFO „Max Valier“ Bozen

School Year 2016/17

Contents

Abbreviation directory

PHP	PHP: Hypertext Preprocessor
PDF	Portable Document Format
FPDF	Free PDF
SQL	Structured value language

1. Introduction

Nowadays a bakery has many different types of recipes. Also different sorts of bread and many other products. Many times, as it is also the case in the bakery of my father, they are written down in different recipe-books or on different sheets of paper. Some of them are also only in the mind of one employee. So it is very hard to have a complete collection of all the recipes in the company. Another problem is that you need every day an other quantity of dough. So you have to calculate anew, how much of every ingredient you need every day. Another problem is that from the year 2017 on, every producer of food in the EU needs to put the nutritional values of their packed products on the package. And they also need to have a list of the nutritional values of the products they sell in an open form, in their store. So it is almost impossible to calculate all these values for all products without a software.

Figure 1 An old recipe book

1.1 The software

To solve, among others, these problems, I have created a PHP web application which allows the user, in my case a bakery, to save all their recipes in a database. They insert all their recipes and can sort them by different categories they belong to. Then they can open the saved recipes and tell the software how many kilograms of dough or how many pieces of bread they need on the given time. Then the software calculates which quantity of each of the ingredients is need. The software calculates also automatically the nutrition-values of all the recipes, inserted in the database based on the nutrition values of their ingredients, which are also saved in the database. So it is possible to print a pdf-page with all the nutrition-values of all the products, but also just for one recipe. Another feature of the software is that it automatically generates the ingredient-list of each recipe with the allergens printed in bold font. They can be used for labels of packed products or for a product list.

1.2 The idea

1.2.1 How I got the idea

Over the past years of summer holidays, I have worked at my fathers bakery as well as in the confectioner. In the beginning I did not know any of the recipes and so I had to ask several times how much of every ingredient I needed, because the recipes were not written down anywhere and only my colleagues knew them. Some others where written down on individual sheets of paper, but not orderly, so that it took a lot of time to find the recipe you searched. And if some baker got ill, it was hard for the others to find out and know all the recipes. So I thought, it would be very useful to have a software, where all the recipes are saved at a local place, ordered and fast and easy to find and access.

1.2.2 The target group

The target group of my software are bakeries and confectioneries, but it could also be used by other companies which make use of many different recipes, for example, also restaurants or producers of food of other categories.

1.3 My motivation

There are many free software projects on the internet. But most of them are made for cooking recipes, and none of them for bakeries and they can´t calculate the nutrition information of the recipes. All bakery-programs available on the market are expensive, for example the software “BackBüro” from the company BÄKO, one of the biggest Austrian s` vendor of products for bakeries. It has approximately the same functions as my software, but it is very complicated to work with. So I wanted to write my own recipe-administration software which should be as easy as possible to work with.

1.4 Time schedules

November 2016: Definition of the database-structure and creation of the database

December 2016: Implementation of the classes, objects and methods in PHP, description of the methods with comments and creating the documentation of the objects and classes

January 2017: Planning and implementation of the user interface

February 2017: Testing the program and debugging it. Eventually implementation of some extra functions.

March and April 2017: Writing the written delivery/test and inserting a lot of recipes as demonstration.

2. The Project

2.1 The project requests

The software should be able to save different categories of recipes. It should also save different ingredients with all their nutrition information that are needed by the law. It should have a login function, so that only a registered user can modify, change, insert or delete recipes. Then you can also change, insert or delete the ingredients, create the price-list and the list with the nutrition information. You also have the possibilities to insert, change or delete the suppliers and the categories. If you aren´t logged in, you can only look at the recipes and calculate the quantities of the recipes and their contents.
The application should only save the ingredients for the dough and not those for the bread, such as, the grains on top of the bread. The recipes are divided in different categories, for example, bread or cakes, so that the confectioner can use it too. Further the program shows a picture of the bread, the mixing time needed by the mixing machine and, if its necessary, the instructions for making it or any extra information needed. Another function should be that the user could make a list in which the recipes are listed in the order that they want/need.
With the application it should also be possible to create a list of all sorts of bread and their ingredients and to print it, so the list can be taken to the shop or used as labels for the packed products, like biscuits. In addition, it should be possible to save which ingredients are in the inventory, their quantity and which deliverer they come from.
For the time the login for multiple users/other bakeries will not be required. If I have some extra time, or after the final exam, I want to implement this too. And I also want to make the function that the software calculates the nutrition information of every bread type. The problem there is, to get the nutrition information of all ingredients into a database. But this is not also absolutely needed at the moment for the program to run.

2.2 Usecase-diagram

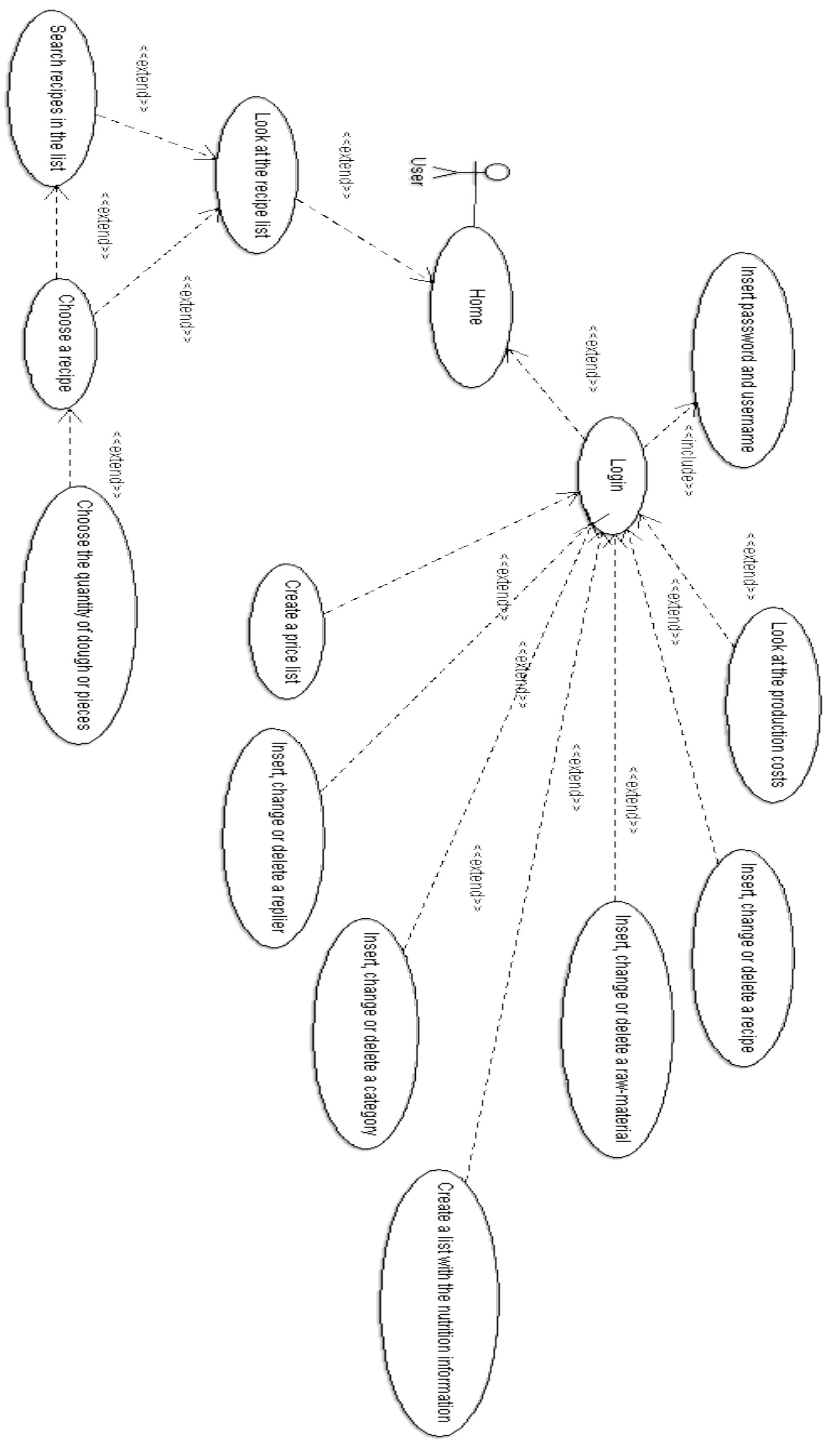

User
Home
Login
Insert password and username
<<include>>
<<extend>>
Look at the recipe list
Search recipes in the list
Choose a recipe
Choose the quantity of dough or pieces
Create a price list
Insert, change or delete a replier
Insert, change or delete a category
Create a list with the nutrition information
Insert, change or delete a raw-material
Insert, change or delete a recipe
Look at the production costs

Figure 2: The UseCase Diagram

2.3 Technologies used

I used the server-side dynamic web programming language PHP for the application, the object classes and the methods that make the connection to the database. In connection with it I used the web language HTML 5 and CSS for the user-interface, and for the database I used the server-software MariaDB with the programming language MySQL. The database was configured with the web-tool phpMyAdmin. The database and the web-application are running on an Apache webserver with the version 2.4.23.

2.4 Security

2.4.1 Authentication

The software uses the HTTP-authentication for the login. So the header() function is used to send an “Authentication Required” message to the browser of the client causing it to pop up a Username/Password input window. Then the inserted values (username and password) can be found in the $_SERVER array. So it is possible to define public and protected areas of the webpage and to realize a login and logout function. But here the security is not very important, because the application is never used by more than one user at the same time and it is only an application within one company, not on the internet, where everyone has access to. The login is only useful to guarantee that not everybody can change or delete the recipes, but only an admin user who has knowledge of the recipes.

2.4.2 SQL Injection

A frequently used technique to steal data from a website is the SQL-Injection. With it the hacker adds some extra SQL-commands to the request to get some extra data from the database, for example with the “UNION” operator from SQL, which is used to connect two select commands. To avoid this SQL injection, the software uses only prepared-statements. So the statement is written to the database just before the user inputs from the text fields get included as parameters. After this, the parameters are sent to the server added to the statement with the bind()-method. This makes a SQL injection impossible.

2.4.3 User passwords

The passwords of the users are saved in the database in form of an MD5-encrypted hash function. This is safer than to save them in decoded form. If someone is able to read the passwords in the database on the server, they are encrypted and not as easy to read and to decrypt.

2.5 Classes and methods

2.5.1 Business object model

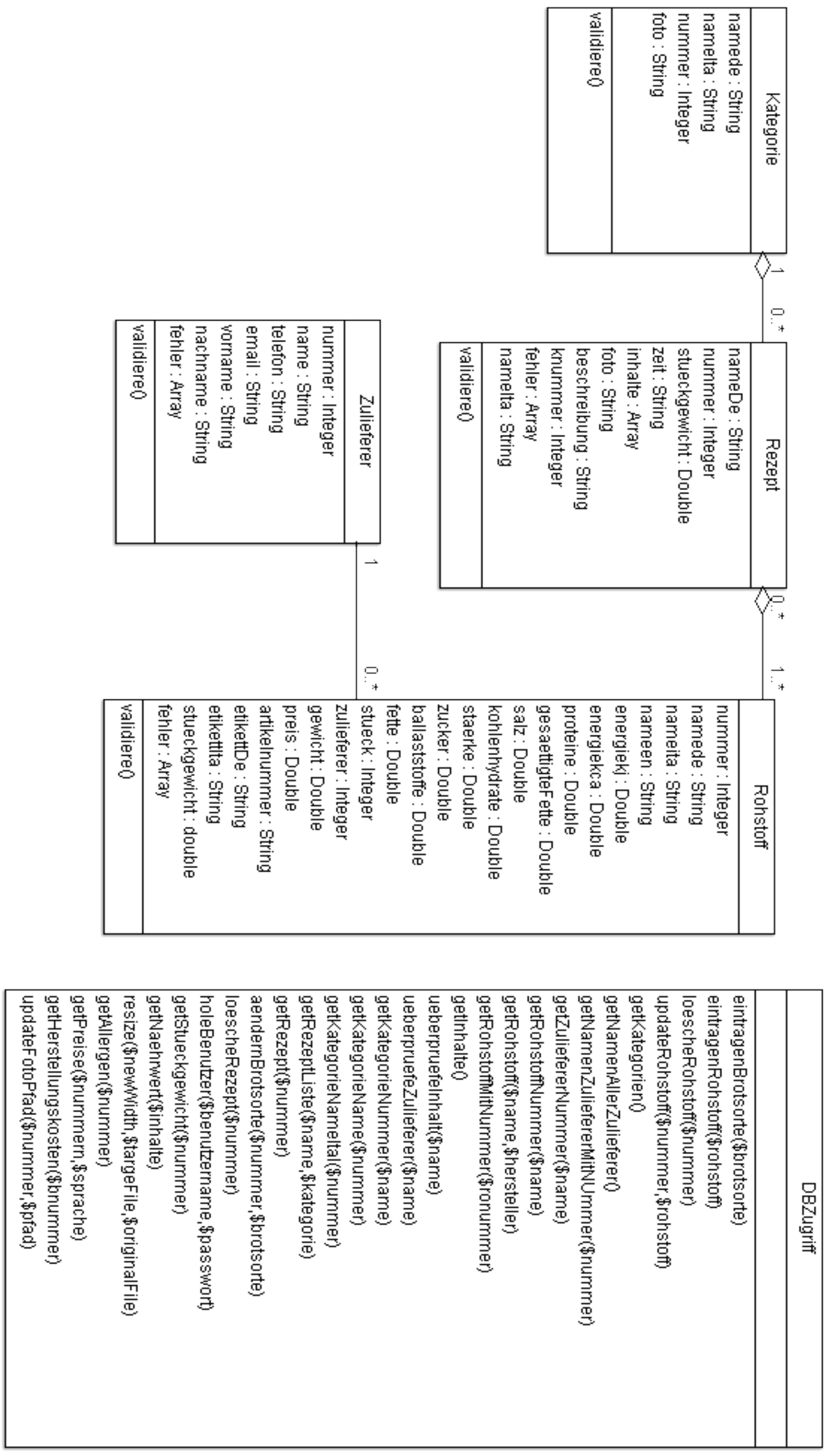


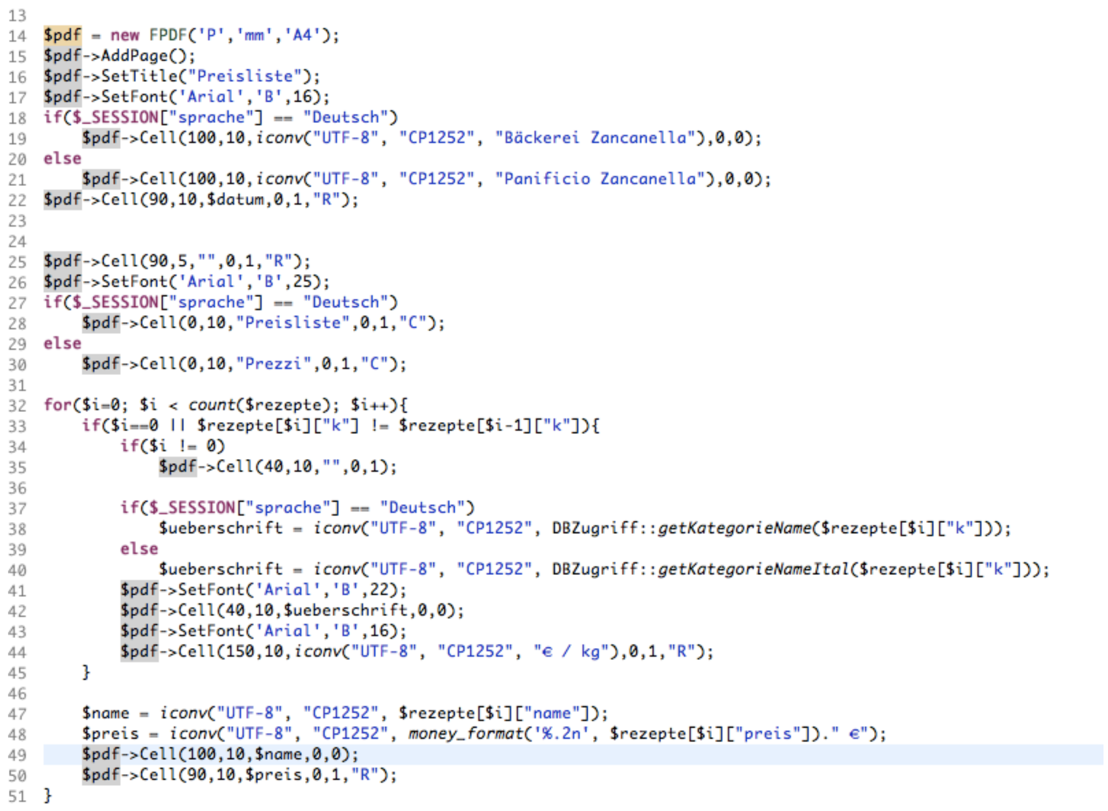
Figure 3: The Business Object Model

2.5.2 The documentation

To see all the information about the classes and their methods, look at the project documentation, created with the tool PHPDocumentor, which allows to generate automatically a documentation of your PHP-classes.
Documentation

2.6 Creation of the PHP-file

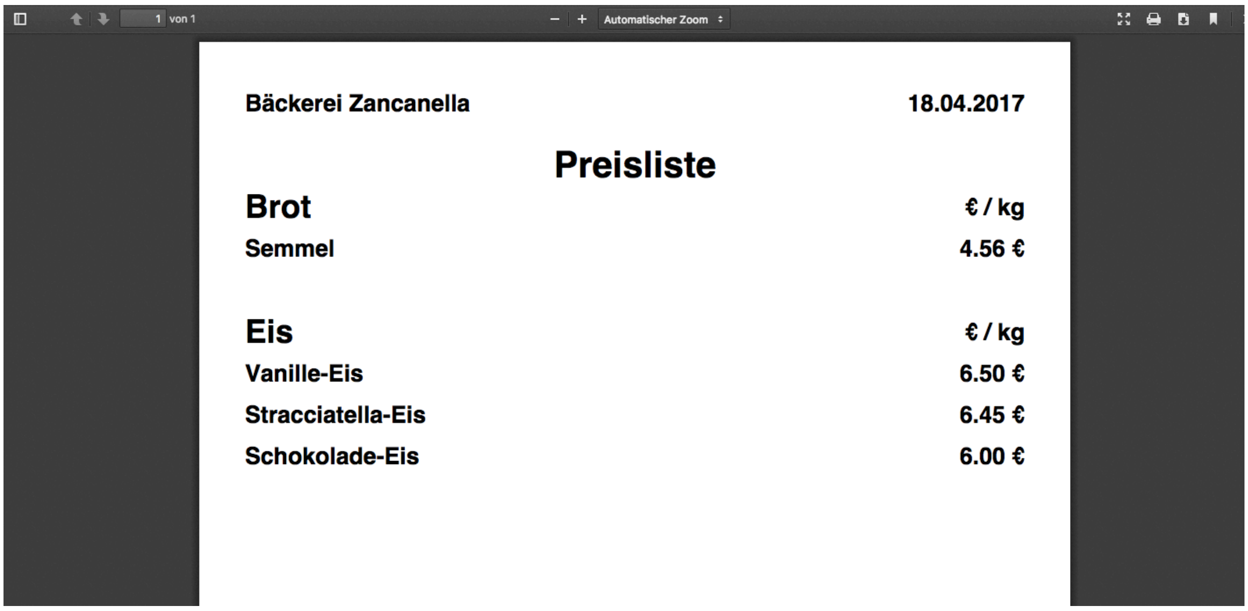
To create the pricelist and the list with the nutrition values in the PDF-format, I used some free available classes written in PHP. The project is called FPDF and the “F” at the beginning stands for “Free”. The classes allow you to generate PDF files with pure PHP. The PDF-site can be handled just as an object and the objects (text, images, positions, ……) can be added by setter-methods. The PHP page appears as output in the browser and is easy to save on your PC or to print.



```php

$pdf = new FPDF('P','mm','A4');
$pdf->AddPage();
$pdf->SetTitle("Preisliste");
$pdf->SetFont('Arial','B',16);
if($_SESSION["sprache"] == "Deutsch")
    $pdf->Cell(100,10,iconv("UTF-8", "CP1252", "Bäckerei Zancanella"),0,0);
else
    $pdf->Cell(100,10,iconv("UTF-8", "CP1252", "Panificio Zancanella"),0,0);
$pdf->Cell(90,10,$datum,0,1,"R");


$pdf->Cell(90,5,"",0,1,"R");
$pdf->SetFont('Arial','B',25);
if($_SESSION["sprache"] == "Deutsch")
    $pdf->Cell(0,10,"Preisliste",0,1,"C");
else
    $pdf->Cell(0,10,"Prezzi",0,1,"C");

for($i=0; $i < count($rezepte); $i++){
    if($i==0 || $rezepte[$i]["k"] != $rezepte[$i-1]["k"]){
        if($i != 0)
            $pdf->Cell(40,10,"",0,1);

        if($_SESSION["sprache"] == "Deutsch")
            $ueberschrift = iconv("UTF-8", "CP1252", DBZugriff::getKategorieName($rezepte[$i]["k"]));
        else
            $ueberschrift = iconv("UTF-8", "CP1252", DBZugriff::getKategorieNameItal($rezepte[$i]["k"]));
        $pdf->SetFont('Arial','B',22);
        $pdf->Cell(40,10,$ueberschrift,0,0);
        $pdf->SetFont('Arial','B',16);
        $pdf->Cell(150,10,iconv("UTF-8", "CP1252", "€ / kg"),0,1,"R");
    }

    $name = iconv("UTF-8", "CP1252", $rezepte[$i]["name"]);
    $preis = iconv("UTF-8", "CP1252", money_format('%.2n', $rezepte[$i]["preis"])." €");
    $pdf->Cell(100,10,$name,0,0);
    $pdf->Cell(90,10,$preis,0,1,"R");
}
```

Figure 4: example of the usage of FPDF

If you generate the pricelist of all inserted bread types and ice cream flavors in the Mozilla Firefox browser, it looks like this example:



Bäckerei Zancanella 18.04.2017

Preisliste

Brot	€ / kg
Semmel	4.56 €

Eis	€ / kg
Vanille-Eis	6.50 €
Stracciatella-Eis	6.45 €
Schokolade-Eis	6.00 €

Figure 5: example for a created PDF price list

2.7 Should/Is comparison

At the moment it is possible to insert, change and delete recipes, supplier, categories and also the different ingredients. Also the login function for an administrator is implemented. The recipes can be saved in different categories and it is just possible to create the PDF pages with the price list and the nutrition information. The webpage is divided in to public area for everyone and a protected area for the administrator. Also the calculation of the quantities for the dough or an exact number of pieces already works fine. But some of the functions mentioned just before are still missing, for example, sorting the recipe list in a random order as the user wants/needs it. I will implement this some time later as an additional feature. I have also not implemented the function to save the quantity in the storage for each ingredient, so that the software could show the baker which ingredient is about to run out, so that the user knows what to order from the suppliers. The problem there is that the software can never know exactly how much of an ingredient is in store if it has not been registered. To implement this function would be very time-consuming and not very efficient, so I decided to omit it. Further I have not implemented the function to save the recipes of more than one user/ more than one bakery in the same database yet, but I will do that in the future.

3. The User Interface

3.1 The main page

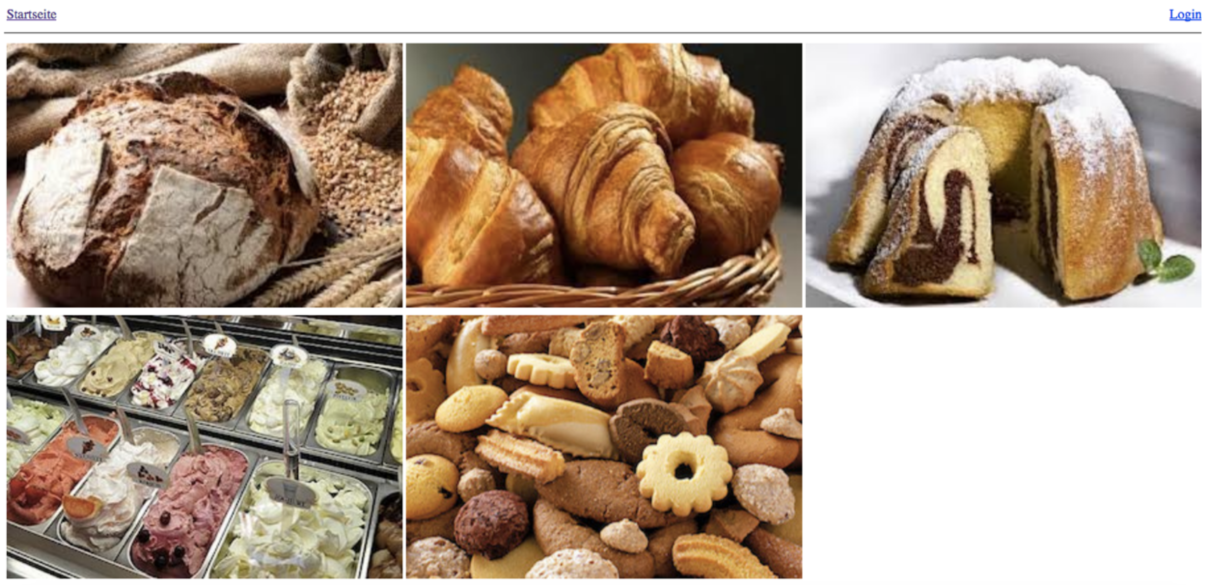


Figure 6: Screenshot of the main page

You will see these images, if you visit the website for the first time. From here the user can log in or can select a category of recipes from which they want to see a list of the inserted recipes. I have decided to keep the design as simply as possible, so that it is easy to work with the software. And it is also not very important that a software for work has a complex user interface. The functionalities to find everything fast and easily are more important than a good interface.

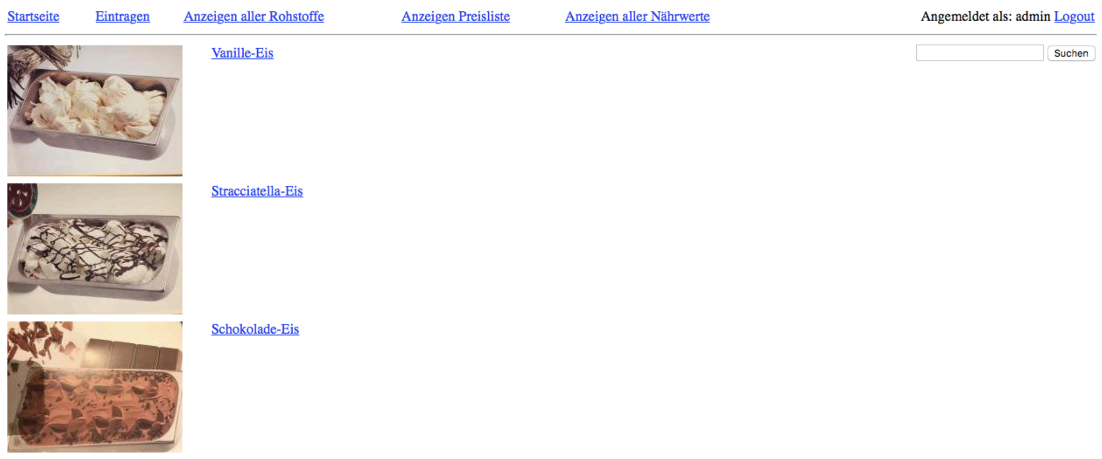
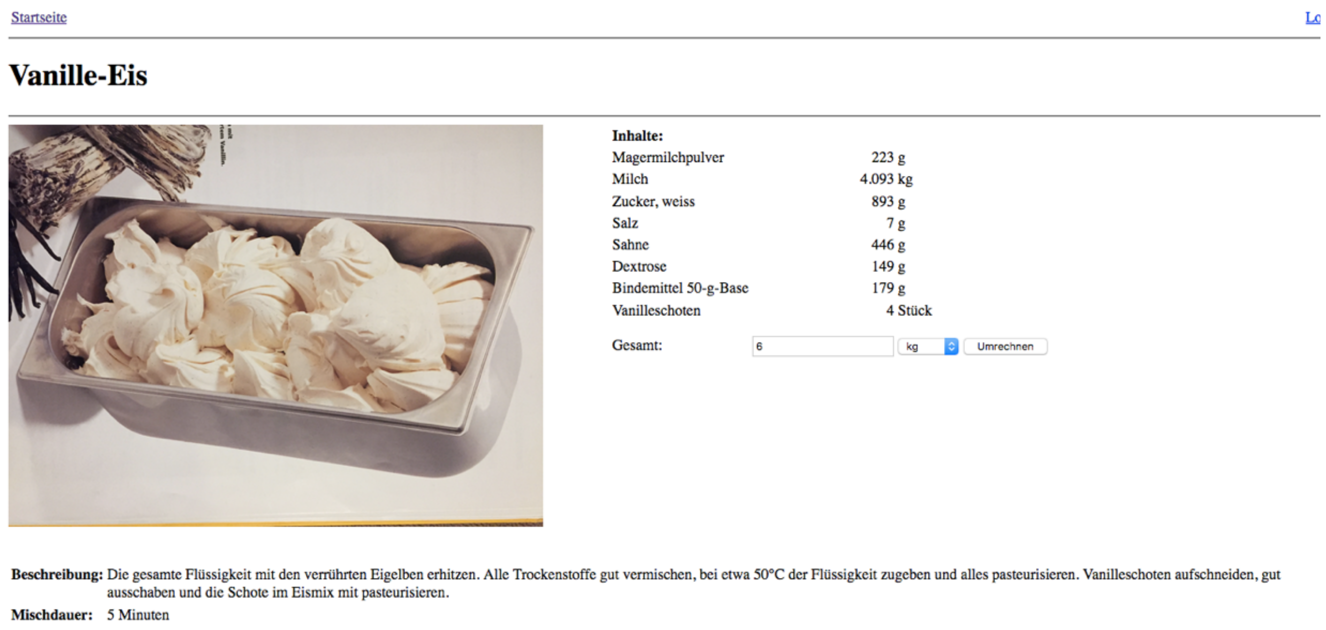


Figure 7: The recipe list if the user is logged in as administrator

3.2 The recipe page



Startseite

Vanille-Eis

Inhalte:	
Magermilchpulver	223 g
Milch	4.093 kg
Zucker, weiss	893 g
Salz	7 g
Sahne	446 g
Dextrose	149 g
Bindemittel 50-g-Base	179 g
Vanilleschoten	4 Stück

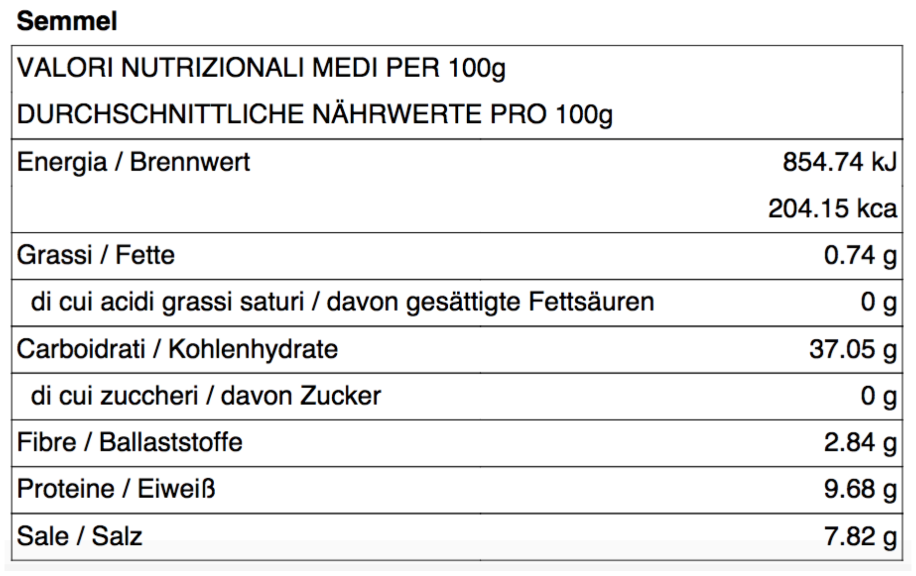
Gesamt: 6 kg Umrechnen

Beschreibung: Die gesamte Flüssigkeit mit den verrührten Eigelben erhitzen. Alle Trockenstoffe gut vermischen, bei etwa 50°C der Flüssigkeit zugeben und alles pasteurisieren. Vanilleschoten aufschneiden, gut ausschaben und die Schote im Eismix mit pasteurisieren.

Mischdauer: 5 Minuten

Figure 8: The recipe page

You will get to this page if you choose a recipe you want to see in detail. Every recipe must have a name, the ingredients and quantities, and it can also include a picture, a description, how to make it and the mixing duration. If you are logged in, you can also see how much the ingredients of a recipe cost for 100g of dough. If the user is logged in, he/she can also modify the recipe, for example change the ingredients or change the price. He/she can also delete the recipe from the database, and there is also the possibility to have a look at the nutrition values of the recipe in use.



Semmel

VALORI NUTRIZIONALI MEDI PER 100g DURCHSCHNITTLICHE NÄHRWERTE PRO 100g	
Energia / Brennwert	854.74 kJ 204.15 kca
Grassi / Fette	0.74 g
di cui acidi grassi saturi / davon gesättigte Fettsäuren	0 g
Carboidrati / Kohlenhydrate	37.05 g
di cui zuccheri / davon Zucker	0 g
Fibre / Ballaststoffe	2.84 g
Proteine / Eiweiß	9.68 g
Sale / Salz	7.82 g

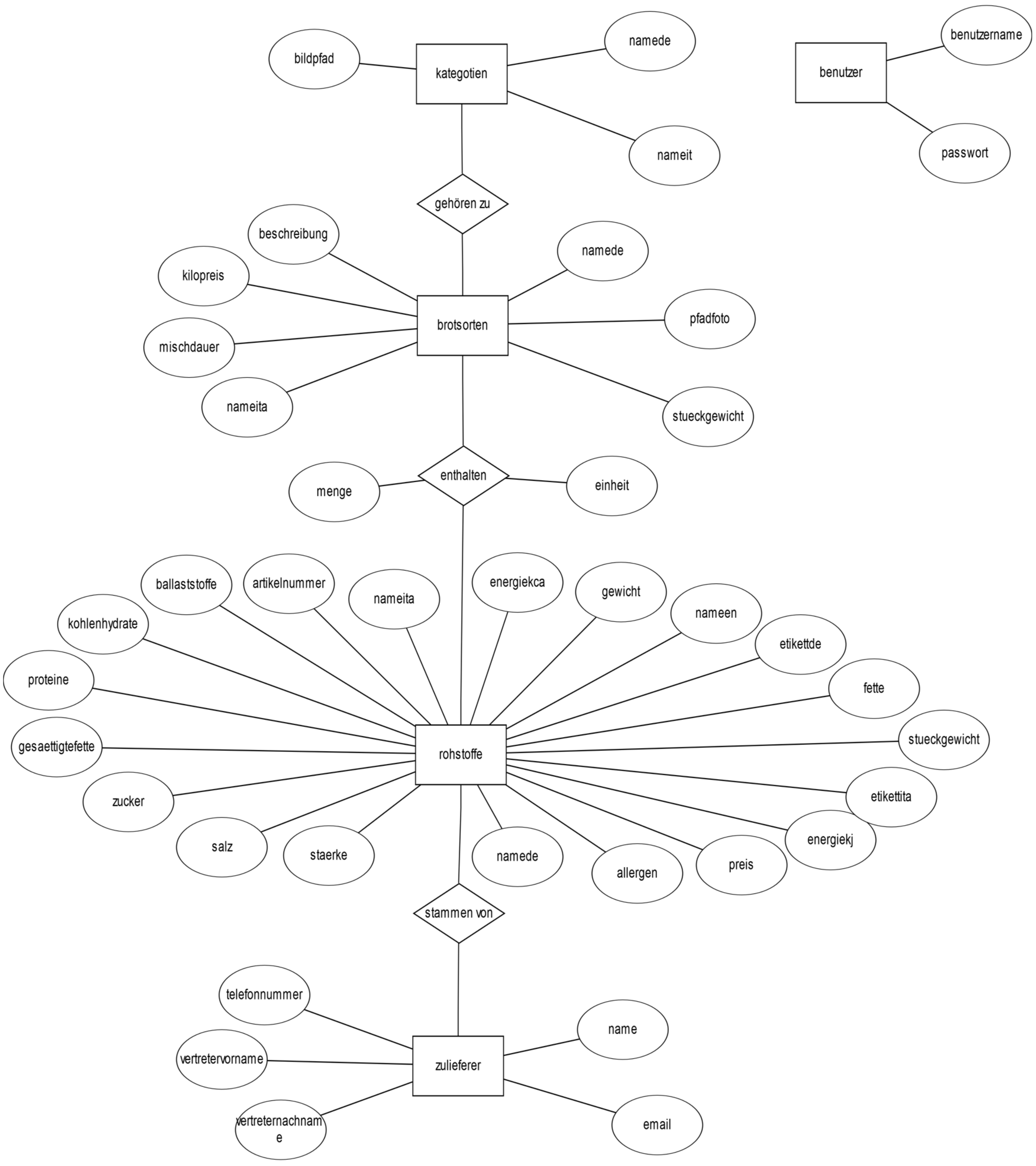
Figure 9: Example of the declaration of the nutrition values

3.3 Organization/Structure of the website

The main page of this web application is the index.php file, which includes all the other pages/interfaces. All links or actions from the forms go to the index.php. There the program checks which number for the id-parameter is set in the request and selects the correct methods and includes the correct user interface. It is also very useful, because this way you have to include all objects just once and further you need to start the session only at the beginning of the index.php file. This file only carries out the controls, whereas most of the output is made in the included script files.

4. The Database

4.1 Entity Relationship Model



benutzer(bbenutzername, bpasswort)

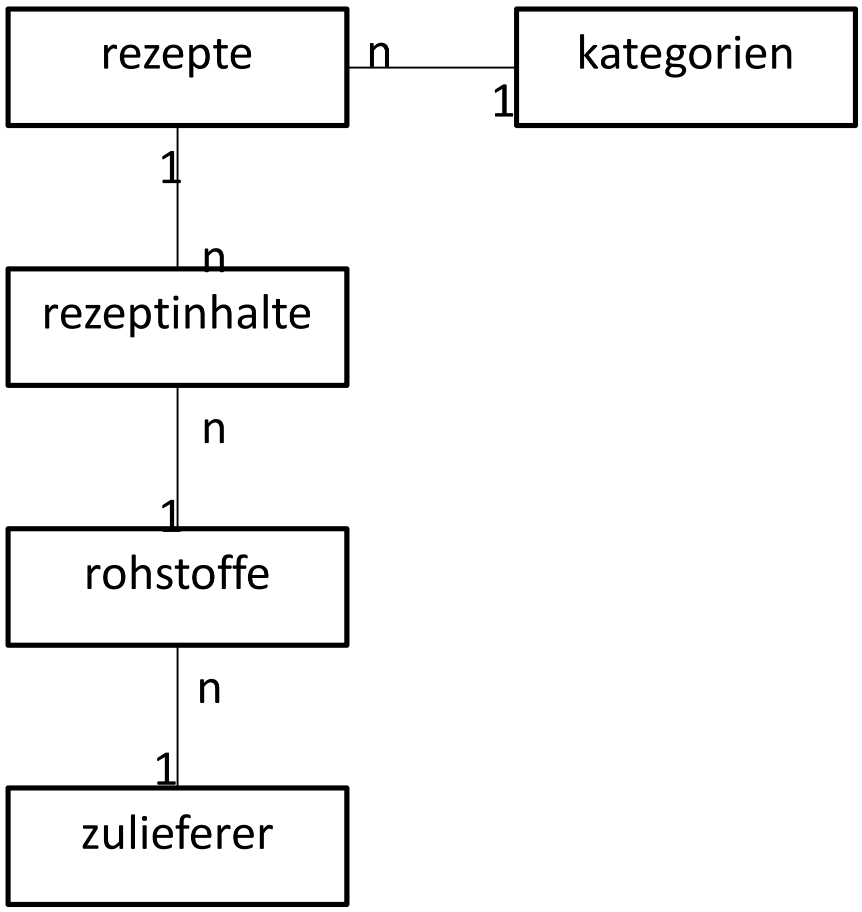
kategorien(knummer, knamede, knameita, kbildpfad)

brotsorten(bnummer, knummer, bnamede, bnameita, bstueckgewicht, bkilopreis, bmischdauer, bbeschreibung, bfoto)

rezeptinhalte(bnummer, ronummer, rimenge, rimasseinheit)

rohstoffe(ronummer, znummer, ronamede, ronameita, ronameen, roallergen, roenergiekj, roenergikca, roproteine, rogesaettigtefette, rofette, rosalz, rokohlenhydrate, rostaerke, rozucker, roballaststoffe, rogewicht, roartikelnummer, ropreis, roetikettde, roetikettita, rostueckgewicht)

zulieferer(znummer, zname, zemail, ztelefonnummer, zvertretervorname, zvertreternachname)



```
CREATE TABLE benutzer(
  bbenutzername varchar(100) NOT NULL,
  bpasswort varchar(100) NOT NULL,
  PRIMARY KEY(bbenutzername)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE kategorien (
  knummer int(11) NOT NULL AUTO_INCREMENT,
  knamede varchar(100) NOT NULL,
  knameIta varchar(200) NOT NULL,
  kbildpfad(300) NOT NULL,
  PRIMARY KEY(knummer)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE brotsorten (
  bnummer int(11) NOT NULL AUTO_INCREMENT,
  knummer int(11) NOT NULL,
  bname varchar(100) NOT NULL,
  bstueckgewicht decimal(10,2),
  bkilopreis double(10,2),
  bmischdauer varchar(10),
  bbeschreibung text,
  bfoto varchar(100),
  bnameita varchar(300),
  PRIMARY KEY (bnummer),
  FOREIGN KEY (knummer) REFERENCES kategorien(knummer)
      ON DELETE RESTRICT ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE zulieferer (
  znummer int(11) NOT NULL AUTO_INCREMENT,
  zname varchar(100),
  zemail varchar(100),
  ztelefon varchar(20),
  zvertretervorname varchar(50),
  zvertreternachname varchar(100),
  PRIMARY KEY(znummer)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE rohstoffe (
  ronummer int(11) NOT NULL AUTO_INCREMENT,
  ronamede varchar(100) NOT NULL,
  ronameita varchar(100) NOT NULL,
  ronameen varchar(100) NOT NULL,
  roallergen tinyint(1) DEFAULT '0',
  roenergiekj double(11,0),
  roenergiekca double(11,0),
  roprotein double(11,0),
  rogesaettigtefette double(11,0),
  rosalz double(11,0) NOT NULL,
  rokohlenhydrate double(11,0),
  rostaerke double(11,0),
  rozucker double(11,0),
  roballaststoffe double(11,0),
  rofette double(10,0),
  rogewicht int(11),
  romengeLager int(11),
  roartikelnummer varchar(100),
  znummer int(11),
  ropreis double NOT NULL,
  roetikettit varchar(300) NOT NULL,
  roetikettde varchar(300) NOT NULL,
  rostueckgewicht double,
  PRIMARY KEY (ronummer),
  FOREIGN KEY (znummer) REFERENCES zulieferer(znummer)
    ON DELETE RESTRICT ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE rezeptinhalte (
  bnummer int(11) NOT NULL,
  ronummer int(11) NOT NULL,
  rimenge int(11) NOT NULL,
  rimasseinheit varchar(50) NOT NULL,
  PRIMARY KEY(bnummer, ronummer),
  FOREIGN KEY(bnummer) REFERENCES brotsorten(bnummer)
        ON DELETE CASCADE ON UPDATE CASCADE,
 FOREIGN KEY(ronummer) REFERENCES rohstoffe(ronummer)
         ON DELETE RESTRICT ON UPDATE RESTRICT
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

4.2 The ingredients information

I obtained the data of the ingredients and their nutrition values from the webpage http://www.naehrwertdaten.ch. It is a webpage of the nutrition-information database from Switzerland. The values can be downloaded in the form of a csv-file for free. I deleted the relevant information and imported the details into my database. I did this with the web tool “PHPMyAdmin”.

5. Conclusion

5.1 Extension possibilities

As already mentioned, in a next step I will first create the possibilities to order the recipe list in a way that the user requires and second the login system fore more than one bakery and multiple users. Another function which could be very useful to implement is to adapt the design of the application to mobile devices, above all tablets, so that it could be used in the bakery on a small device, which would be much handier and easier to use during work as compared to a computer.

6. Appendix

6.1 Source code

Link to the source code

6.2 Project documentation

Link to the documentation

Bibliography

http://php.net/manual/en/features.http-auth.php

http://codefactory.esy.es/php/create-thumbnail-and-resize-image/

https://www.phpdoc.org

http://www.naehrwertdaten.ch/request?xml=MessageData&xml=MetaData&xsl=Start&lan=de&pageKey=Start

http://www.radgmbh.de/wp-content/uploads/2011/03/Fotolia_5466993_XS.jpg

Informatics school documents on moodle

Eidesstattliche Erklärung

„Ich versichere, dass ich die vorstehende Arbeit selbständig angefertigt und mich fremder Hilfe nicht bedient habe. Alle Stellen, die wörtlich oder sinngemäß veröffentlichtem oder nicht veröffentlichtem Schrifttum entnommen sind, habe ich als solche kenntlich gemacht.“