

Project Databases Report

Pizzeria Ordering System

Teachers : Jos de Bruin
 Dr. Hamideh Afsarmanesh
Assistant: Simon Msanjila

Students:
Tyrone Vriesde Tvriesde@science.uva.nl 0588067
Tony Cheong W.T.T.Cheong@student.uva.nl 0477400
Ivan Joghi IvanJoghi@gmail.com 0564974

Executive Summary

A custom pizza online website was created for customers of a pizzeria to create their orders online. The pizzeria has been taking orders by phone for a long time and they wanted to add the option for customers to create orders online.

Existing webshop solutions involves putting products in categories. Such a division is not suitable for ordering custom pizzas. There is a need to allow users to pick ingredients for a pizza, ability to configure multiple custom pizzas and the ability to order other products besides pizzas.

Our solution separates the ordering of products in Non pizza products and Custom Pizzas which consists of a selection of ingredients for pizzas. The website runs on Open Source software, it is written in PHP and uses MySQL as the database manager.

The website allows customers to create custom pizzas, pick non pizza products and enter delivery information. Employees can process accept or deny orders, and update the status as processing orders continues. The customer will be kept up to date by e-mail notification when the order status changes. The administrator can add and edit ingredients, non pizza products and he can also add Employee accounts so the employees can log in to process the orders.

Table of Contents

Executive Summary	2
Table of Contents	3
Chapter 1: Problem description	4
1.1 Introduction.....	4
Goal.....	4
Existing solutions.....	4
Our solution	4
Technical constraints	4
1.2 Use Cases	5
Ordering	5
Order processing	5
Administration	5
1.3 System specification	5
Functionality specification.....	5
Technical specification	6
1.4 Requirements	7
Users	7
Functional requirements.....	7
Non-functional requirements	8
Chapter 2: Approach	9
2.1 Technical Approach	10
Chapter 3: Results	11
3.1 Architecture.....	11
Information Architecture	11
System Architecture.....	13
3.2 GUI Specification	13
3.3 Function implementation queries.....	16
Chapter 4: Evaluation	25
Chapter 5: Conclusion.....	33
Appendix A:.....	34
User manual	34

Chapter 1: Problem description

1.1 Introduction

A pizzeria specialized in custom made pizzas is currently taking orders by phone. The current system where the customer calls the pizzeria takes time of employees to answer the phone and is more work consuming than necessary. They want to allow customers to customize and order their pizzas online. The pizzeria also aims to increase the sales, due to the easy to use order online website. The system will give the employees more time to “work” rather than to accept orders by phone, also the potential increase in customers are enough reason for the pizzeria to accept the change (website where customers can order their customized pizzas).

Goal

Our goal is to deliver a database with a user interface (website) where customers can select various ingredients for their own pizza and place their order. The order will be sent to the “kitchen” where the pizza will be made. The focus is to create an “easy to use” website, which will allow a first time customer to complete their order with ease.

Existing solutions

There are many different web based ordering systems. Those systems do not offer all the functionality that is needed for a pizzeria. Ordering systems usually allow people to add products and separate them in different categories and subcategories. Pizzeria sells pizzas, and most of them also offer clients to customize their own pizzas by picking their own ingredients. At the same time they offer other products that do not need customization like the side orders. That does not fit into the general category-subcategory differentiation.

Our solution

Our solution is to make an ordering system that separates ordering pizzas from ordering the side dishes (non-pizza products) in an intuitive way. The ordering system provides the user with three tabs: one for customizing pizzas, one for side orders, and one for delivery details. Customers can switch to any tab any time. Hence the system will decrease workload of the employees and benefit the pizzeria due to the database / information system. Information will be stored in the system and can be viewed at any time. The system will be able to guide a user through the website and make them complete their pizza order. When they are done with filling in all information regarding their order they can complete the order to send it to the pizzeria.

Technical constraints

The constraints of this case are the network connectivity which is required and a website located on the internet. Customers will have to be able to customize their pizzas. However there is a limit to the customization of the pizza, technical constraints, functions of the database and user interface. The system should make use of open-source software, since the pizzeria does not want an expensive system.

1.2 Use Cases

Three main use cases are identified: *ordering*, *order processing* and *administration*. The following paragraphs describe each use case in more detail.

Ordering

The first use case is ordering. To order, a potential customer must first know what he can choose from (“What’s on the menu?”). For ordering custom pizzas a customer will have to know the ingredients he can choose from and how choosing ingredients will influence the total price. Customers pick ingredients for their pizza and sometimes they want a double portion of specific ingredients. When a customer is done choosing his pizza, he may want to order several pizzas of the same configuration. A customer may also want to order non-pizza side orders, such as drinks and salads. Also the customer will have to provide his name, address and telephone number. A customer can also pick another delivery date and time if they wish for the order to be delivered on a later date and time. Customers can order up to two weeks in advance.

Order processing

When an order has been confirmed by a customer order processing begins. Throughout the order processing the order status will be updated and the customer will be notified. New orders will show up in a list for the employees in the kitchen. They need to state which products are ordered and which ingredients are chosen for the custom pizzas. The kitchen employees have to look at the order and decide if they can prepare the order or not. When the decision is made, the order status will change and the customer will be notified. After preparing the order the order status will be updated and the delivery employees will take over. They will look up the name and address, and deliver the order. The customer will also be notified when the order is prepared and ready to be delivered. When delivery has been completed or failed, the order status will be changed once more.

Administration

Custom pizzas are made by selecting several ingredients, the list of available ingredients and their prices are administrated by an employee, for instance the manager. Administration includes added, editing and deleting ingredients. Besides the administration of ingredients, the non-pizza side orders must also be administrated. Administrators also need to have logs of previous orders.

1.3 System specification

Functionality specification

The functionality specification is closely related to the use cases. The three main functions are *ordering*, *order processing* and *administration*. A description of each of the functionality will follow. It includes the output of the system and the input it expects from the users.

Ordering involves customizing pizzas, selecting products and entering customer information. The system will have to provide the user with data about the ingredients and the non-pizza products such as name and price. Users will provide an order which

consists of custom pizzas by selecting ingredients, non-pizza products, and customer information like name, address and telephone number. For every custom pizza and non-pizza product users can also specify an amount if they wish to order more than one of them.

Order processing is involved in preparing and delivering orders. The system needs to provide the kitchen employees with a list of pending orders. Of each order the ordered products and pizzas must be shown, together with the amount of each product. For each custom pizza the selected ingredients must also be shown. The kitchen employees will first view the order and decide if they are going to process the order or not. The decision will be entered by changing the status of the order to the respective status. An e-mail will be sent to customer to keep him informed of the ordering status. After the order is prepared, the order will be passed to the delivery employees who will use the system to retrieve the customer details.. The status of the order will change to “being delivered” and an new e-mail will be sent to the customer. When the order is delivered, the delivery employee can mark the order as “delivered”, or in case he failed to deliver it he will make it as “failed to deliver”.

Administration includes adding, editing and deleting available ingredients, non-pizza products and employees and viewing previous orders. The system will be able to show a list of ingredients and a list of products. For each ingredient and non-pizza product a name, price and picture can be specified. The system also needs a function to show previous orders.

Technical specification

The system is made up of three layers. At the top there is the GUI (Graphical User Interface) layer, the middle layer is the storage and query manager, the bottom layer is the underlying database.

GUI layer

The GUI layer allows users to access the system. All the functionalities of the system must be available through the GUI. There are two separate GUI's. One is for the customers to create orders and one is for employees for processing orders and administration purposes. With scripts, user input will be used to invoke queries from the storage and query manager layer to provide the user with various pages. The GUI should prevent input errors and in case of errors that could not be prevented, provide clear error messages.

Nowadays people have usernames and passwords for a lot of websites and services. It is not practical to have users to remember information for a pizzeria. It is much easier for customers to type in their name and address than to have to remember the username and password. Therefore, customers do not have accounts to log on to. However customer information will be stored into the system to allow employees view previous orders by customers.

The GUI for employees is on a separate URL. Users need a username and a password to gain access to the system. The administrator has access to add users to the system to give them access.

Storage and query manager layer

The storage and query manager layer is responsible for information storage, retrieval, authorizations and error checking. This layer allows selecting, adding, updating, deleting entities and relations in the database by using different queries. Some

instructions are limited to users with authorization, such as deleting data from the database which should only be allowed by the administrator.

Database layer

The database layer contains all the data of entities and their relationships.

1.4 Requirements

This section describes different types of users of the system in 3.3.1. In the following paragraph 3.3.2, the functional requirements for each type of user are listed. 3.3.3 Lists the non-functional requirements of the system.

Users

Three types of users should be able to use the system: customer, employee and administrator. Customers are users who visit the website and can create orders by customizing pizzas, selecting products and entering customer details. Employees are the group of users that work with the ordering system on a daily basis. Employees will have their own accounts to log on to. They are the ones responsible for processing orders. Since Customer users do not need a log in, employees who process telephone orders can use the system as a Customer and enter the telephone order directly into the system as they take the order from the calling customer.

The administrator, or super user, has the ultimate control of the system, he can add, change or delete ingredients and products, as well as add, change, or delete employee accounts.

Functional requirements

1 Customers

- 1.1 The user must be able to create a new order.
- 1.2 The user must be able to customize a pizza by:
 - 1.2.1 The user must be able to view a list of available ingredients.
 - 1.2.2 The user must be able to add an ingredient to a custom pizza
 - 1.2.3 The user must be able to remove an ingredient from a custom pizza
 - 1.2.4 The user must be able to get graphical feedback from selecting ingredients. A photo of a pizza will contain the newly selected ingredient combined with previous selected ingredients.
- 1.3 The user must be able to add a custom pizza to an order.
- 1.4 The user must be able to view a list of available non-pizza products.
- 1.5 The user must be able to add non-pizza products to an order.
- 1.6 The user must be able to see a list of custom pizzas and non-pizza products that are added to the order.
- 1.7 The user must be able to change the amount of a custom pizza.
- 1.8 The user must be able to change the amount of a non-pizza product.
- 1.9 The user must be able to delete a custom pizza from an order.
- 1.10 The user must be able to delete a non-pizza product from an order.
- 1.11 The user must be able to see the total price of an order.
- 1.12 The user must be able to choose a delivery date and time that is up to two weeks ahead.
- 1.13 The user must be able to add the name and address of the customer.
- 1.14 The user must be able to clear the current order to start a new one.

1.15 The user must be able to confirm the order.

2 Employees

2.1 The employee must be able to log in and out.

2.2 The employee must be able to view a list of available orders and their custom pizzas.

2.3 The employee must be able to mark orders as “prepared”.

2.4 The employee must be able to mark order as “delivered”

2.5 The employee must be able to mark order as “failure to deliver”

2.6 Only users with respective rights (employee) must be able to use all these “Employees” features.

3 Administrators

3.1 The administrator must be able to log in and out.

3.2 The administrator must be able to add/delete/edit orders.

3.3 The administrator must be able to add/delete/edit ingredients.

3.4 The administrator must be able to add/delete/edit non-pizza products.

3.5 The administrator must be able to add/delete/edit other users.

3.6 The administrator must be able to view an order log.

3.7 Only users with respective rights (administrators) must be able to use all these “Administrators” features.

Non-functional requirements

As an operational requirement, the system will run as a database with a website as user interface. As performance requirement the system must be accessible 24 hours a day, seven days a week. Due to the nature of the system as an ordering website, the system must have a low response time, preferably shorter than second, with a maximum of five seconds. The exception is viewing order logs which could have a higher response time (of seconds) as the log increases in size over time.

Due to the low complexity of the system, no problems with response time are expected. Customers who visit the website to order will get a sessionID for their visit, which is used to identify them while using the system. For every action they take, a timestamp is stored. From time to time a service on the server will scan sessionID's and timestamps. SessionID's which have not been active for more than three hours will be deleted along with the corresponding ordering information.

Chapter 2: Approach

Our solution to make the pizzeria more efficient will be a website to create orders online, which should be come with a good user interface. Also it should be “fast” as in the customer will be able to order his pizza in a matter of minutes. We stress the importance of “easy to use” user interface here because some customers will leave if the website is too complicated. In the case of a website where a user goes to for personal interest a less user friendly interface is not a big problem since he will try and try to get the information due to his interest in the site.

To create the website we will first analyze the current processes of the pizzeria. This will be recorded in the use cases. From the use cases we derive the system specification which describes what the system should be able to do. It will contain the needed functionalities and the technical specification. Then we will analyze the different users and list their functional requirements and non-functional requirements for the system in detail.

With all the specification and requirements we will create an information model using an E-R diagram to describe the entities and the relations between the entities. From the E-R diagram we will derive a relational schema which will be used for the final database design. The E-R diagram will be made in Visio.

GUI prototypes will follow using screen mockups in the form of images. These mockups will be created using Photoshop.

To support implementation, a list of functions will be created describing what actions these functions need to perform. The list will also include the SQL statements that are needed to perform these actions.

During implementation, test scripts will be created. After implementation is complete, the test scripts will be run by a few people to detect implementation errors. The project will end when the implementation errors are fixed.

Project planning and Workload distribution

	Tyrone	Ivan	Tony	Deadline Week	Actual Delivery Week*
Use cases	2	2	1	37	38
System specification	2	2	1	37	38
Requirements	2	2	1	37	39
E-R diagram	1	2	2	38	38
Relational schema	1	2	2	39	39
GUI prototypes	2	2	1	39	40
Function list	2	1	2	40	40
Implementation	1	2	2	40	41
Test scripts	2	1	2	41	41
Testing	2	1	2	41	41
Manual	2	1	2	42	42

1 = Main Responsibility 2 = Takes Part in

* Tracked for reflection purposes

2.1 Technical Approach

During the process of selecting the tools and methods to be used during the process, we took a look at our current expertise and the requirements of the project.

As a programming paradigm, ASP.NET would have been the best choice at first, but the fact that this is not an open source environment, forced us to start looking for a different approach. Because the database was going to be a mysql driven database, php showed to be the right choice. Php has an excellent integration with mysql. Because php is an open source system, there are a huge amount of hosts available.

For managing the mysql database, phpmyadmin was used. This was the logical choice because it is free, and it integrates so well. Creating the code, the program Macromedia Dreamweaver was used. This is a multipurpose xhtml editor, supporting a lot of different syntax languages. We used this program because we already had experience using it.

The system is divided into two main parts. The smaller administration area, and the larger customer area. Both parts have their own folders, for example: localhost/pizza is the folder containing the customer site, and localhost/pizza/admin is the folder containing the admin page.

Both parts have been designed around the index.php page. This page is dynamically being changed based on values in the POST and the html Query String. Different types of pages are being included when needed. We divide the include pages by, library pages and include pages. Library pages are pages which only execute code, but do not write html code. Include pages execute code and write html.

Finally, some of our limited graphic design has been done using Macromedia Fireworks. This because we already had good experience using this tool combined with Macromedia Dreamweaver.

To sum it up:

Database Managing System: MySQL

Tool to access the database: phpmyadmin

Programming paradigm: PHP

Tool used to edit php: Macromedia Dreamweaver

Tool used for graphics design: Macromedia Fireworks

Chapter 3: Results

Contains the information architecture in the form of ERD and relational schema. 3.5 includes the GUI specification and finally 3.6 contains a list of functions with implementation steps.

3.1 Architecture

Information Architecture

This paragraph describes the information architecture for the database system. The data model consists of an Entity Relationship Diagram (ERD) and relational schemas

ERD

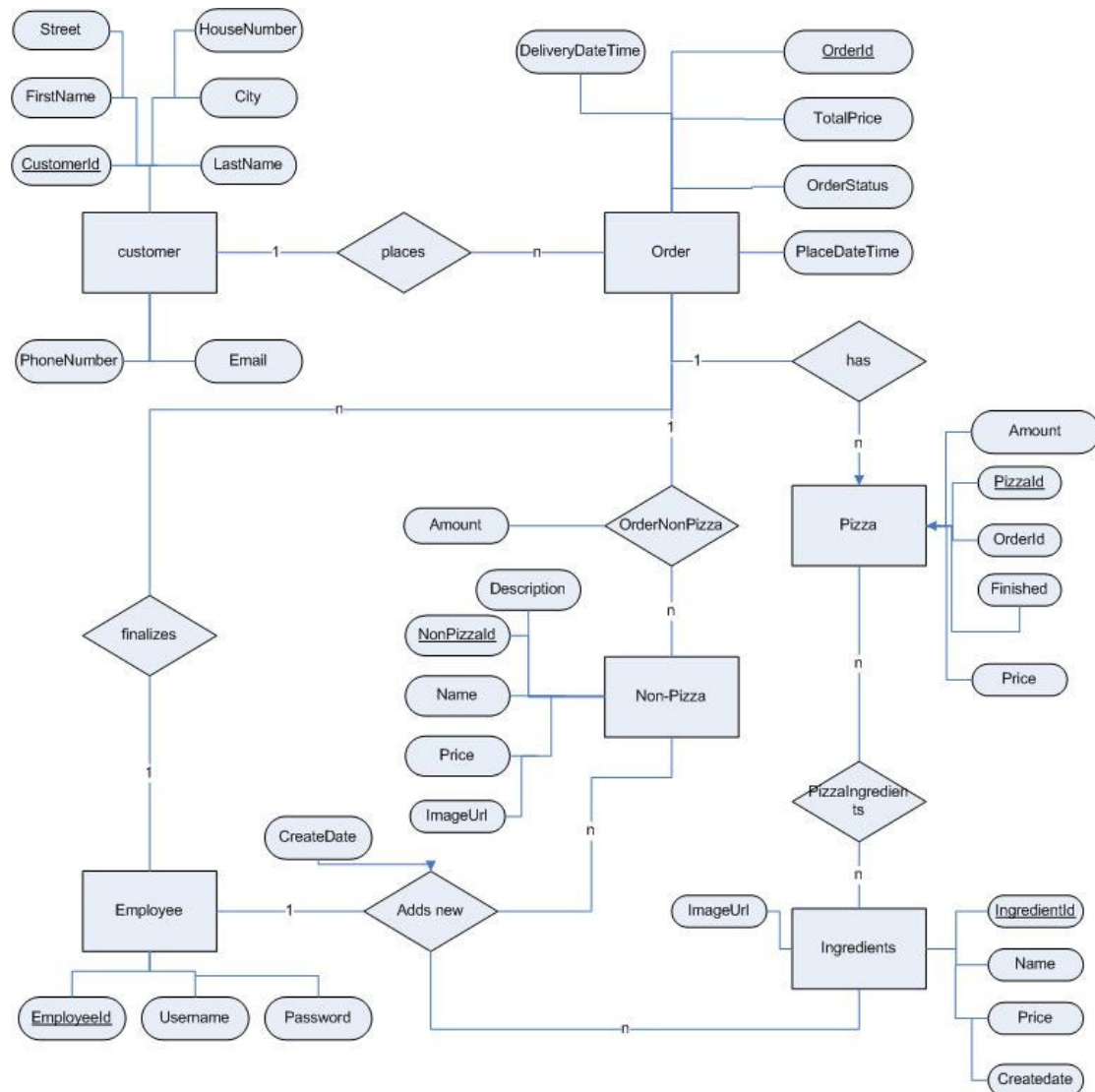


Diagram 1: ERD

We chose for a separate Non-Pizza and Pizza table, because it is far more clear to program. If we would combine both the tables, we would get confusing situations,

because Pizza records are being created by the customers, and non-pizza records are being created by the administrators. The attributes might be very similar but a pizza record is always connected to a bunch of ingredient records, where this is not the case for the non-pizza records. The price of a pizza is based on the individual ingredient prices. Upon completing the pizza, the price is being inserted in the Pizza table.

Diagram 1 shows the Entity-Relationship Diagram (ERD). In this paragraph every relation between entities is described.

Customer --- Places --- Order

A customer can place several orders; the customer entity holds attributes describing properties concerning the customer. The order entity holds attributes such as OrderStatus to track the status of the order (new/prepared/delivered/failed to deliver).

Employee --- finalizes --- Order

An employee is responsible for the multiple or one order. By tracking this, it is possible to look up which employee finalized which order.

Employee --- Adds New --- Non-Pizza

Through this relationship it is possible to track which non-pizza product has been added to the database by which employee.

Employee --- Adds New --- Ingredients

Through this relationship we track which ingredient has been added by which employee.

Pizza --- PizzaIngredients --- Ingredients

This relation describes which ingredients belong to the pizza creations (created by customers during their shopping process). A pizza can have 1 or multiple ingredients. It is impossible for a pizza to have zero ingredients.

Order --- OrderNonPizza --- Non-Pizza

This tracks which non-Pizza products have been selected by the customer during the shopping process. An Order can have 1, multiple or none Non-Pizza products.

Order --- Has --- Pizza

This relation tracks the pizza's which belong to an order created by a customer.

Relational schema

Diagram 2 shows the relational schemas of the database. It is derived from the ERD.

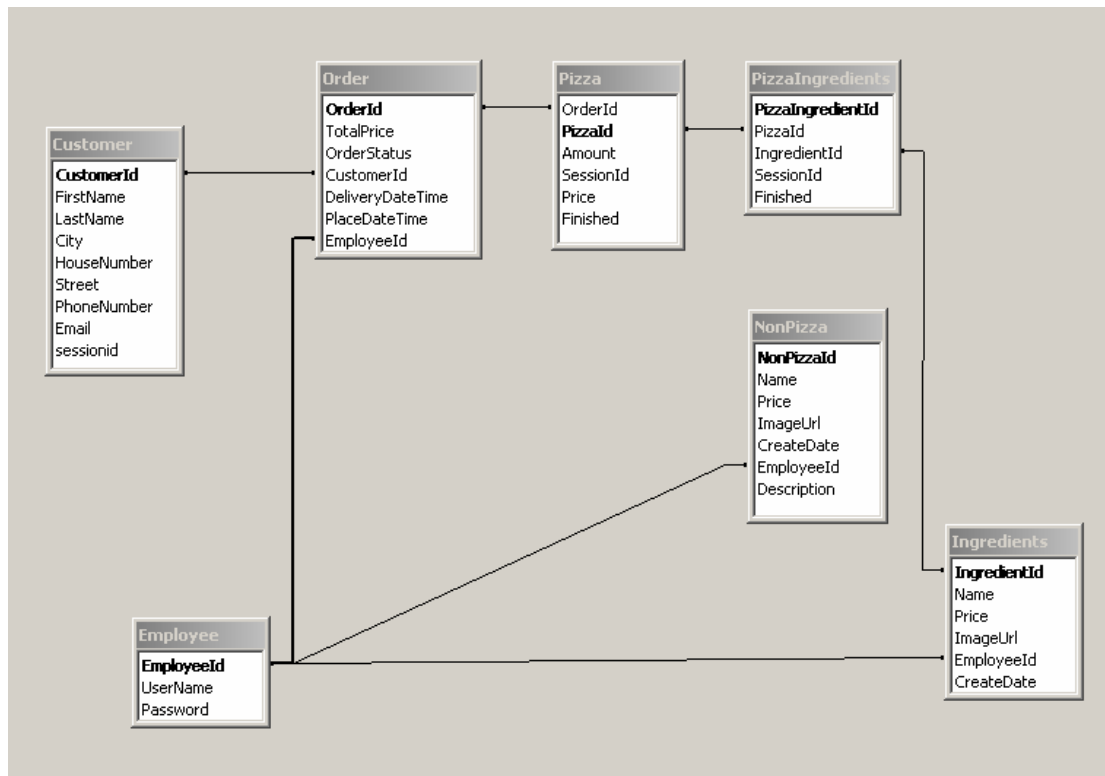


Diagram 2: Relational schemas

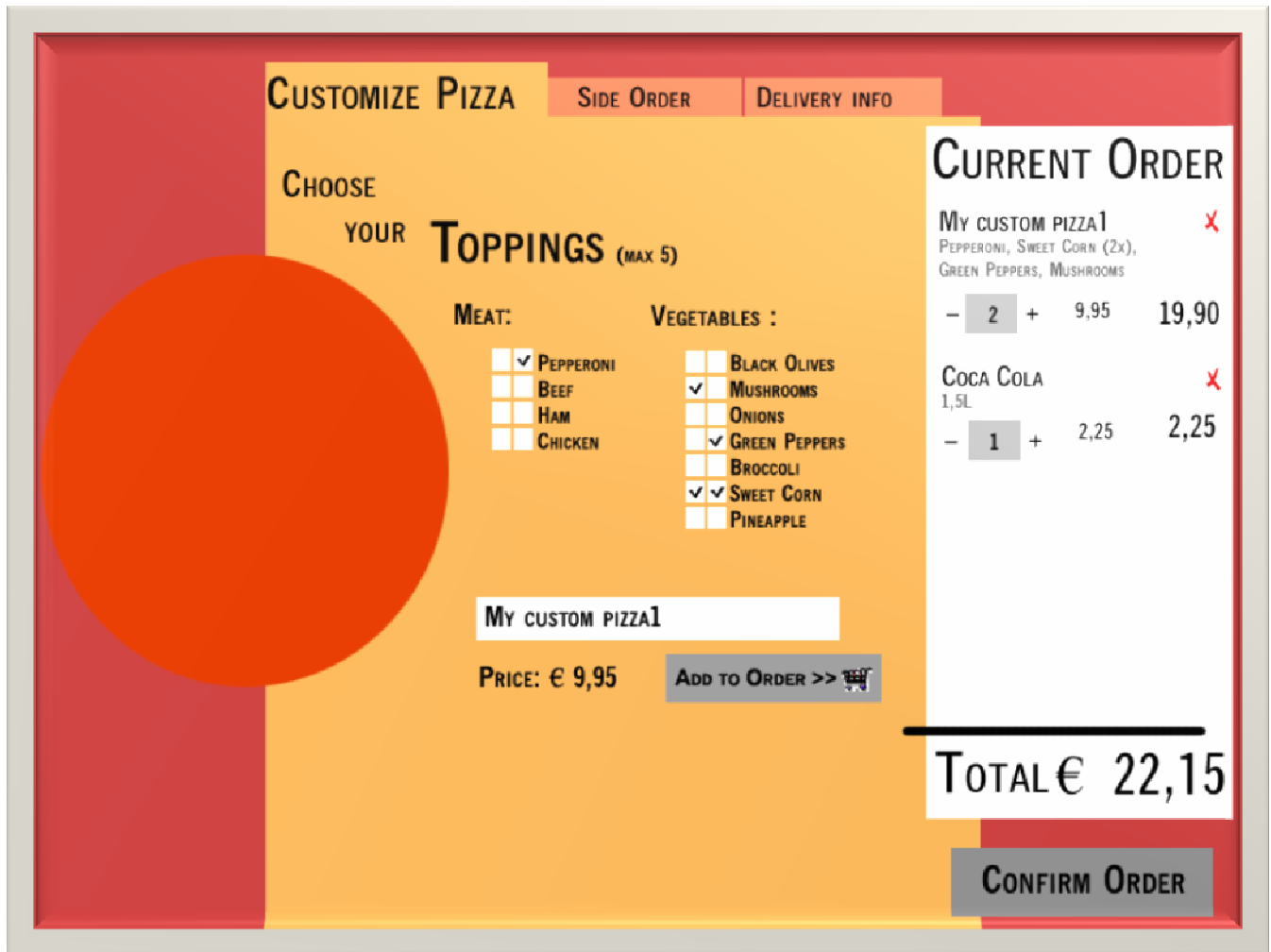
As visible in the Relational schema, a new attribute has seen the light. This attribute is named Sessionid. The sessionid is a random id which is assigned to a user when he enters the website, during the visit he will have the same sessionid. This enables, multiple users to place orders at the same time. If we would not have used a sessionid, the users could see and change the process of other customers, causing a lot of inconvenience.

System Architecture

As one of the constraints of the project, the database system will run on Open Source software. A powerful solution for a database system such as the pizza ordering website is LAMP. LAMP is a software bundle which enables running dynamic websites with an underlying database. This acronym stands for Linux, Apache, MySQL and PHP. A server will run on Linux as the OS. Apache is the web server. The database will run on MySQL and PHP serves as the programming language. This combination is cheap and provides all the functionality which is needed. This combination is also offered by a large portion of the hosting companies, making it easy to find a place to put the system on.

3.2 GUI Specification

The GUI is web-based and will use web-based technologies such as HTML, JavaScript and PHP in combination with images to display pages. The following paragraphs will describe several pages from the system.



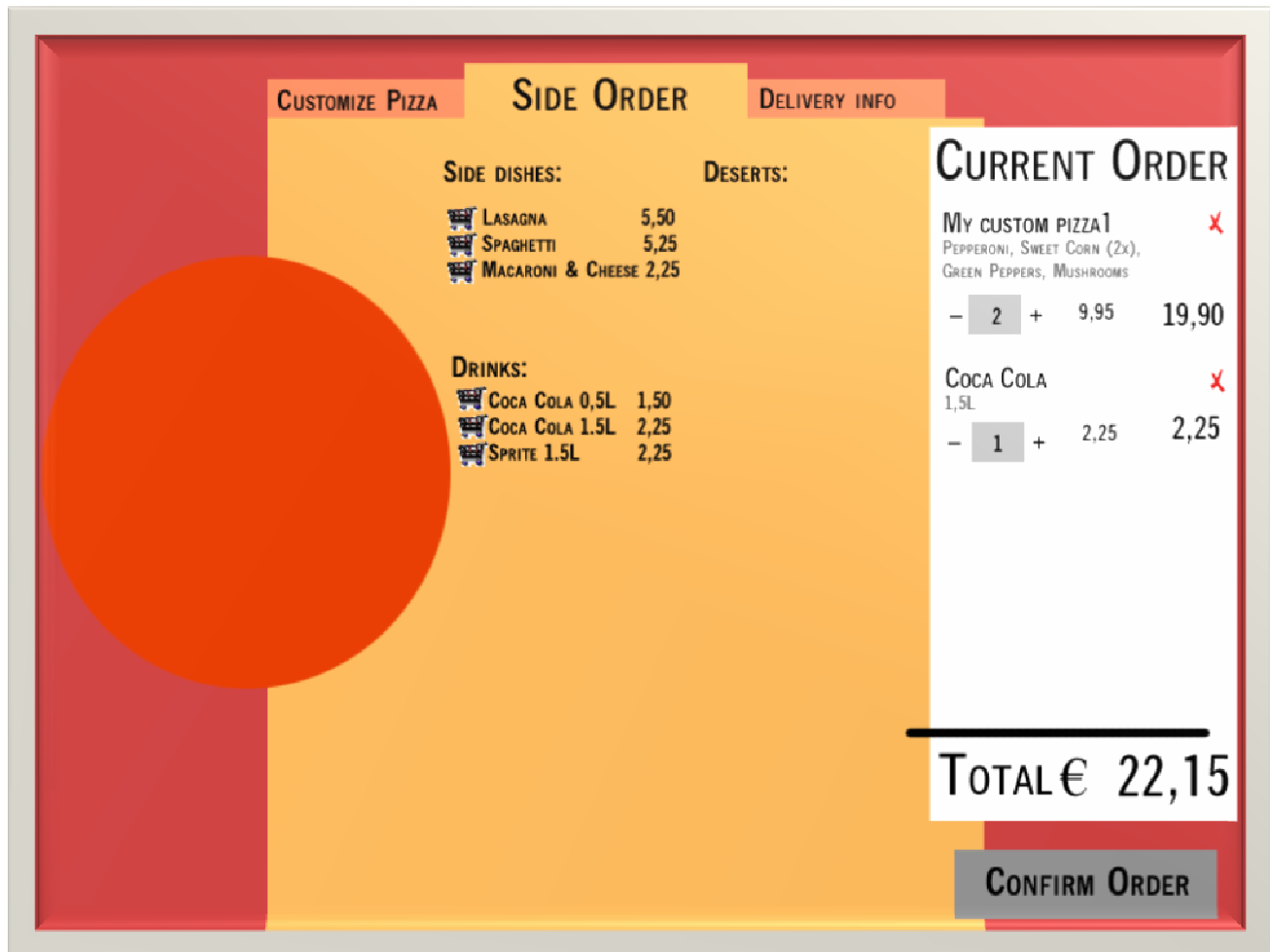
Screenshot 1: Customize Pizza screen

Screenshot 1 shows the first page for customers. In the middle of the screen are all the available toppings for custom pizzas. Next to each ingredient are two checkboxes. This allows users to select zero, one or double portions of the ingredient which corresponds to how pizzas are generally ordered. A big pizza is shown on the left. The image will be updated every time the user selects or removes an ingredient. The image will show the selected ingredients on the pizza and it looks like a photo of an actual pizza. The realism will increase the appetite of the user when he is creating his order. The total price will update every time an ingredient is selected. There is a button that allows the user to create a new pizza.

On the right there is a list of currently ordered pizzas and products. For every item the user can adjust the amount by clicking on the – or + buttons and users can also delete the item from the order by clicking on the red cross.

On top of the screen there is a tab-like interface. The first tab is “Customize Pizza”, the middle tab is “Side Orders” for drinks, salads and deserts, the last tab is “Delivery Information” where delivery information such as name, address and time is entered. A tabbed interface allows users to browse through several in the order they want to. An example: a user adds pizzas, picks side orders, enters customer information and then decides he wants to order another pizza. He can do it by clicking on the

Customiza Pizza tab and add a new pizza. In a wizard interface the user would need to click Next and Back buttons to find the page he was looking for. Another alternative was to have a menu of three hyperlinks under each other. A tabbed interface is the better choice, with only three tabs it shows clearly that there are three different pages corresponding to the three steps of ordering.



Screenshot 2: Side Order

Screenshot 2 shows the Side Order tab. This page shows non-pizza products ordered by category. Clicking on the shopping cart will add the product to the order list. Just like the previous page, tabs are shown on top of the page and the order list is shown on the right.

CUSTOMIZE PIZZA | **SIDE ORDER** | **DELIVERY INFO**

CUSTOMER INFORMATION

FIRST NAME

LAST NAME

STREETNAME

HOUSE NUMBER

CITY

PHONENUMBER

IF YOU WANT YOUR ORDER DELIVERED ON A DIFFERENT DATE AND TIME, PLEASE ENTER THE DESIRED DATE AND TIME.
WE DELIVER 24/7.

CURRENT ORDER

My Custom Pizza1 ✗
PEPPERONI, SWEET CORN (2x),
GREEN PEPPERS, MUSHROOMS

- 2 + 9,95 19,90

Coca Cola ✗
1,5L

- 1 + 2,25 2,25

TOTAL € 22,15

CONFIRM ORDER

Screenshot 3: Delivery Information

Screenshot 3 shows the delivery information tab. Customers can fill in their first name, last name, street name, house number, city and phone number. Optionally if they want the order to be delivered on another date or time, they can specify it too.

On all pages, under the order list there is a clearly visible button available to confirm the order. However, to confirm an order, there must be products on the order list and customer information must be filled in.

When “Confirm Order” is pressed and there are no products ordered, a text message will notify the user that he has to order something first.

When “Confirm Order” is pressed and the customer information page is not filled in, the system will show the Delivery Info page for users to fill in.

When “Confirm Order” is pressed and all the information is entered, a confirmation page will show the delivery information and the order details. The user can click on “OK” to confirm the order or “Cancel” to continue ordering.

3.3 Function implementation queries

This paragraph contains a table which list all the needed functions with steps that the scripts needs to take and what DB steps are required. Another table contains the actual DB queries.

#	Name	Goal	Description	Steps	DB steps
1	Create new order	create_new_order	The user has opened the start page of the ordering system and is ready to create a new order.	<ul style="list-style-type: none"> - User opens the start page of the ordering system - Create new custom pizza 	<ul style="list-style-type: none"> - New sessionId - New Pizza.PizzaId - New ContainsNonPizza.ContainsNonPizzaId - Set Pizza.amount to "1"
2	View ingredients	view_ingredients	The customize pizza page shows a list of available ingredients for the pizzas.	<ul style="list-style-type: none"> - Query database for data - Show the list of ingredients 	<ul style="list-style-type: none"> - Select "IngredientId", "Name", "Price" from "Ingredients" table
3	Select ingredient	select_ingredient	The user can add an ingredient to the custom pizza.	<ul style="list-style-type: none"> - Add selected ingredient to the custom pizza 	<ul style="list-style-type: none"> - New PizzaIngredients.PizzaingredientId - Store Ingredients.IngredientId as PizzaIngredients.IngredientId - Store Pizza.PizzaId as PizzaIngredients.PizzaId
4	remove_ingredient	remove_ingredient	The user can remove an ingredient from the custom pizza by unticking a tick box next to the ingredient.. A user can only untick a ticked boxed.	<ul style="list-style-type: none"> - User Clicks on a ticked tick box next to ingredient - Remove ingredient from the custom pizza 	<ul style="list-style-type: none"> - Remove record where PizzaIngredients.IngredientId = current IngredientId and Pizza.PizzaId = PizzaIngredients.PizzaId
5	Show_pizza	show_pizza	A photo of a pizza will be shown in the customize pizza page to reflect the custom pizza. Adding ingredients will update the photo with a new photo with all the ingredients on	<ul style="list-style-type: none"> - Get URL's from ingredients - Show selected ingredients 	<ul style="list-style-type: none"> - Select Ingredients.ImageUrl where PizzaIngredients.PizzaId is current PizzaId and PizzaIngredients.IngredientId = Ingredients.IngredientId

			top of it.		
7	Add_pizza	add_pizza	On the customize pizza page, the user can add the pizza, as it is customized, to the order. When the user starts creating another pizza, the previous pizza is being completed.	- Add pizza	update pizza set Finished = 'T' where sessionid = 'sessionid' and finished = 'F'"
8	View_nonpizza	view_nonpizza	A list of all available non-pizza will be shown	- Query database for data - Show non-pizza products	- Select "NonPizzaId", "Name", "Price" from "NonPizza" table
9	Add_nonpizza	add_nonpizza	The user add the product to the order. If it is already added, it will increase the amount by 1 with each click.	- Query database for data - Add Nonpizza product to the order or increase the amount	- Select ContainsNonPizzaId where sessionId is current sessionId and ContainsNonPizza. NonPizzaId = NonPizza.NonPizzaId - If record exist, increase amount by 1 - Else create such a record with amount 1. .
10	View_order	view_order	From all the pages the current order can be viewed. It contains a list of pizzas, non-pizza products, the amount of each product the price of each product and the total price of the order.	- Query database for data - Show ordered products - Calculate total price	- Select all from Pizza where SessionId is current SessionId. - Select Price from Ingredients where Ingredients.IngredientId = PizzaIngredients.IngredientId and PizzaIngredients.PizzaId = Pizza.PizzaId
11	Change Amount pizza	change_amount_pizza	When a pizza is added to the order, the amount can be changed.	- Change Pizza amount	- Update Pizza.Amount where Pizza.SessionId is current SessionId
12	Change Amount nonpizza	change_amount_nonpizza	When a non-pizza product	- Change NonPizza amount	- Update ContainsNonPizza.Amount where

			is added to the order, the amount can be changed.		ContainsNonPizza.SessionId is current SessionId
13	Delete pizza	delete_pizza	After a pizza is added to the order, the pizza can be deleted.	- Delete Pizza	delete from pizza where pizzaid not in(select pizzaid from pizzaingredients) and sessionid = 'sessionid'
14	Delete nonpizza	delete_nonpizza	After a non-pizza product is added to the order, it can be deleted	- Confirm deletion - Delete NonPizza product from the Order	Delete record where ContainsNonPizza.SessionId is current SessionId and NonPizzaId is to be deleted product.
15	Total price	total_price	The total price is part of view_order, showing the total price of the order.	- Query database - Calculate price for each pizza - Calculate total price for Pizza's and NonPizza's	- For each PizzaId where the SessionId is the current SessionId, get all the Ingredients and the price of each ingredient. - For each NonPizzaId with the current SessionId, get the NonPizza.Price
16	Delivery date	delivery_date	"Delivery details". This opens the delivery details page where the user can specify a delivery date if the user wants the order delivered on another day or time	- Store date - Store time	Right after the creation of a customer record, we will create an order record. insert into `order` (`TotalPrice`, `OrderStatus`, `CustomerId`, `PlaceDateTime`) values ('totalprice', 'desired delivery date time', 'new', 'customerid', 'currentdatetime')
17	Add customer info	add_customerinfo	The user fills in his name, address, telephone number	- Store name - Store address - Store telephone number	insert into customer (Firstname, LastName, City, HouseNumber, Street, PhoneNumber, sessionid, Email) values ('firstname', 'lastname', 'city', 'housenumber', 'street', 'phonenumber', 'sessionid', 'email')
18	Clear_order	clear_order	The user can clear the order and start a new one.	- Delete current order	- Delete PizzaIngredients records where PizzaId = Pizza.PizzaId and sessionid is current sessionid. - Delete Pizza records where sessionid is current sessionid

19	Confirm order	confirm_order	The user can confirm the order when he is done ordering.	- Store order information	This is actually the procedure of inserting a customer record and an order record.
20	Employee views orders	View orders	Employee logs in the system to view all orders	-Employee logs in -Employee goes to orders -Employee views orders	- Select "UserName", "Password" from table Employee Select Ingredients.IngredientId, Ingredients.Name from Ingredients table -Select NonPizzaId, Name from NonPizza table -Select OrderId, DeliveryDateTime, PlaceDateTime, PizzaId, NonPizzaId from Order, Pizza, ContainersNonPizza tables Where OrderStatus = "new", Order.OrderId = Pizza.OrderId, Order.OrderId = ContainsNonPizza.OrderId For every PizzaId found in the previous data: -Select IngredientId from PizzaIngredients, Pizza Where Pizza.PizzaId = PizzaIngredients.PizzaId
21	Employee marks orders	Marks orders	Employee marks orders as "done"	-Employee logs in -Employee goes to orders -Employee marks orders as "done"	- Update Order.OrderStatus = "Done"
22	Employee deletes order from the database	Delete order	Employee deletes order from the database	-Employee logs in -Employee goes to orders -Employee deletes orders	- Delete Order, Pizza, ContainsPizza, PizzaIngredients Where OrderId is the selected Order to be deleted
23	Employee edits order from the database	Edit order	Employee edits order from the database	-Employee logs in -Employee goes to orders -Employee edits orders	- Select Order, Pizza, ContainsPizza, PizzaIngredients Where OrderId is the selected Order to be edited Update Order, Pizza, ContainsPizza, PizzaIngredients Where OrderId is the selected Order to be edited
24	Employee views	View ingredients	Employee views	-Employee logs in -Employee goes to	- Select Ingredients

	ingredients from the database		ingredients from the database	ingredients -Employee views ingredients	
25	Employee adds ingredients to the database	Add ingredients	Employee adds ingredients to the database	Employee logs in -Employee goes to ingredients -Employee adds ingredients	- Add new Ingredient
26	Employee deletes ingredients from the database	Deletes ingredients	Employee deletes ingredients from the database	Employee logs in -Employee goes to ingredients -Employee deletes ingredients	- Delete Ingredient Where IngredientId is Ingredient to be deleted
27	Employee edits ingredients from the database	Edits ingredients	Employee edits ingredients from the database	Employee logs in -Employee goes to ingredients -Employee edits ingredients	- Update Ingredient Where IngredientId is Ingredient to be edited
28	Employee views non pizza products from the database	Views non pizza products	Employee adds non pizza products to the database	Employee logs in -Employee goes to non pizza products -Employee views non pizza products	- Select NonPizza
29	Employee adds non pizza products to the database	Adds non pizza products	Employee adds non pizza products to the database	-Employee logs in -Employee goes to non pizza products	

				-Employee adds non pizza products	- Add NonPizza
30	Employee deletes non pizza products from the database	Deletes non pizza products	Employee deletes non pizza products from the database	-Employee logs in -Employee goes to non pizza products -Employee deletes non pizza products	- Delete NonPizza Where NonPizzaId is NonPizza product to be deleted
31	Employee edits non pizza products from the database	Edits non pizza products	Employee Edits non pizza products from the database	-Employee logs in -Employee goes to non pizza products -Employee edit non pizza products	- Edit NonPizza Where NonPizzaId is NonPizza product to be edit
32	Employee views other users from the database	View users from the database	Employee view users from the database	-Employee logs in -Employee goes to non user section -Employee views users from the database	- Select Employee
33	Employee must be able to view order log	View order log	Employee views order log from the database	-Employee logs in -Employee goes to view order log -Employee views order log	- Select Order

Admin Queries

#	Query Goal	Query Code	Query Parameters
1	Insertion of a new Ingredient	Insert into ingredient (name, price, imageurl, employee) values ('name', 'price', 'imageurl', 'employeeid')	<u>Name</u> : Ingredient name <u>Price</u> : Ingredient Price <u>ImageUrl</u> : The path to the image belonging to the ingredient <u>Employeeid</u> : The employee which added this ingredient to the database
2	Check if login is good or bad	select count(*) as Count from Employee where Username = 'username' and Password = 'password'	Username: the desired username Password:the desired password
3	Register at site	insert into Employee (Username, Password) values 'username' and 'password'	Username: the desired username Password:the desired password
4	Check order	SELECT count(*) FROM `Order` where OrderStatus = 'New'	Orderstatus: The completed orders
#	Query Goal	Query Code	Query Parameters
5	Insert new nonpizza	insert into nonpizza (name, price, imageurl, employeeid) values ('name','price','imageurl','employeeid')	<u>Name</u> : nonpizza name <u>Price</u> : nonpizza Price <u>ImageUrl</u> : The path to the image belonging to the nonpizza <u>Employeeid</u> : The employee which added this nonpizza to the database
6	Insert new ingredient	insert into ingredients (name, price, imageurl, employeeid) values ('name','price','imageurl','employeeid')	<u>Name</u> : Ingredient name <u>Price</u> : Ingredient Price <u>ImageUrl</u> : The path to the image belonging to the ingredient <u>Employeeid</u> : The employee which added this ingredient to the database
7	modify nonpizza	update nonpizza (name, price, imageurl, employeeid) values ('name','price','imageurl','employeeid')	<u>Name</u> : nonpizza name <u>Price</u> : nonpizza Price <u>ImageUrl</u> : The path to the image belonging to the nonpizza <u>Employeeid</u> : The employee which added this nonpizza to the database
8	modify ingredient	update ingredients (name, price, imageurl, employeeid) values ('name','price','imageurl','employeeid')	<u>Name</u> : Ingredient name <u>Price</u> : Ingredient Price <u>ImageUrl</u> : The path to the image belonging to the ingredient <u>Employeeid</u> : The employee which added this ingredient to the database
9	Delete nonpizza	delete from nonpizza where nonpizzaid =nonpizzaid	Nonpizzaid: the id that is given to a nonpizza product
10	Delete ingredient	delete from ingredients where ingredientid = ingredientid	Ingredientid: the id that is given to a ingredient added to the database

Customer Queries:

#	Query Goal	Query Code	Query Parameters
1	Make the pizzaid match to the current pizza → Add ingredient to pizza	select distinct(p.pizzaid) as pizzaid from pizza p inner join pizzaingredients p2 on p.pizzaid = p2.pizzaid where p.sessionid = sessionid and p.finished	<u>pizzaid</u> : pizza name <u>pizza</u> : pizza name of current pizza in making <u>pizzaingredients</u> : The ingredients belonging to the customization of pizza's <u>sessionid</u> : The session which added this pizza/order to the database
2	add non pizza product to nonpizzaorder table → Add non pizza products to order	insert into ordernonpizza(NonPizzaId, Amount, SessionId) values(sessionid)	<u>NonPizzaId</u> : The nonpizza id for the product (example coca cola drink) <u>Amount</u> : The price of the product selected <u>SessionId</u> : The session which the customer is working/ordering
3	Remove pizza from order list	delete from pizza where pizzaid not in(select pizzaid from pizzaingredients) and sessionid = sessionid	<u>pizzaid</u> : pizza name <u>pizza</u> : pizza name of current pizza in making <u>SessionId</u> : The session which the customer is working/ordering
4	Show created pizza's of user session	select pizzaid, price, amount from pizza where sessionid=sessionid	<u>pizzaid</u> : pizza name <u>pizza</u> : pizza name of current pizza in making <u>Price</u> : Price of products/pizza's <u>Amount</u> : Number of products/pizza's <u>SessionId</u> : The session which the customer is working/ordering
5	Total price of order	select sum(n.price * o.amount) as price from ordernonpizza o inner join nonpizza n on o.nonpizzaid = n.nonpizzaid where o.sessionid=sessionid	<u>sum</u> : total amount which is the price of products times the amount <u>Price</u> : nonpizza price or pizza price <u>NonPizzaId</u> : The nonpizza id for the product (example coca cola drink) <u>PizzaId</u> : The pizza id for the product (example customized pizza with x ingredients) <u>SessionId</u> : The session which the customer is working/ordering

Chapter 4: Evaluation

Performance and usability tests:

The tests used come from www.usabilitynet.org

“Performance testing is a rigorous usability evaluation of a working system under realistic conditions to identify usability problems and to compare measures such as success rate, task time and user satisfaction with requirements.”

The benefits from the tests are:

Major usability problems are identified that may not be revealed by less formal testing, including problems related to the specific skills and expectations of the users. Measures can be obtained for the users' effectiveness, efficiency and satisfaction.

The performance of the system depends on the user's connection to the internet, the query and system performance overall is excellent. All test scenario's were completed with success. The overall complexity of the system is quite low, with that in mind the system was made so first time users can make use of all the options which are available. The customer is able to complete the order that he/she wants, which includes “pizza, non-pizza products and the contact information for the order”.

TESTING TEMPLATES:

Ingredient Management

Add and Change Ingredient

Steps	Actions
1.	Add ingredient
2.	Logout
3.	Login Different user
4.	Change Ingredient
V	Are the changes being processed correctly, and does the new record show the different user as employee who changed it? YES
R	The change here means the add or change of any ingredient

V = Validation Point;R = Remarks

Delete Ingredients

Steps	Actions
1.	Delete ingredient
2.	Logout
3.	Login Different user
4.	Delete other Ingredient
V	Are the changes being processed correctly, and does the new record show the different user as employee who changed it? YES
R	The change here means deletion of any ingredient.

V = Validation Point;R = Remarks

Modify Ingredients

Steps	Actions
1.	Modify ingredient
2.	Logout
3.	Login Different user
4.	Modify other Ingredient
V	Are the changes being processed correctly, and does the new record show the different user as employee who changed it? YES
R	The change here means modification to the ingredient.

V = Validation Point;R = Remarks

Non-pizza Management

View Non-Pizza Product

Steps	Actions
1.	View Non-Pizza Product
2.	Logout
3.	Login Different user
4.	View other Non-Pizza Product
V	Are the changes being processed correctly, and does the new record show non-pizza products? YES
R	The non-pizza product which was viewed remains the same

V = Validation Point;R = Remarks

Add Non-Pizza Product

Steps	Actions
1.	Add Non-Pizza Product
2.	Logout
3.	Login Different user
4.	Add other Non-Pizza Product
V	Are the changes being processed correctly, and does the new record show the other or new non-pizza products? YES
R	

V = Validation Point;R = Remarks

Modify Non-Pizza Product

Steps	Actions
1.	Modify Non-Pizza Product
2.	Logout
3.	Login Different user
4.	modify other Non-Pizza Product
V	Are the changes being processed correctly, and does the new record show the different user as employee who changed it? YES
R	The modification here can be the adjustment in the price

V = Validation Point;R = Remarks

Order Management

View Order

Steps	Actions
1.	View Order
2.	Logout
3.	Login Different user
4.	View another Order
V	Are the changes being processed correctly, and does the new record show the order as employee who changed it? YES
R	

V = Validation Point;R = Remarks

Removal Management

Delete Pizza

Steps	Actions
1.	Delete Pizza
2.	Logout
3.	Login Different user
4.	Delete other Pizza
V	Are the changes being processed correctly, and does the new record show the updated pizza list? YES
R	The deleted item should not be in the system anymore

V = Validation Point;R = Remarks

Delete Non-Pizza

Steps	Actions
1.	Delete Non-Pizza
2.	Logout
3.	Login Different user
4.	Delete another Non-Pizza
V	Are the changes being processed correctly, and does the new record show the updated non nonpizza list? YES
R	The deleted item should not be in the system anymore

V = Validation Point;R = Remarks

Information Management

Delivery Date

Steps	Actions
1.	Check Delivery Date
2.	Logout
3.	Login Different user
4.	Check the Delivery Date of another Order
V	Are the changes being processed correctly, and does the new record show the different Delivery Date of the new order/another order? YES
R	

V = Validation Point;R = Remarks

Add Customer info

Steps	Actions
1.	Add Customer Info
2.	Logout
3.	Login Different user
4.	Add other Customer info
V	Are the changes being processed correctly, and does the new record show the different user as the correct customer information? YES
R	A new user will have a clean form to fill in (always)

V = Validation Point;R = Remarks

Clear Order

Steps	Actions
1.	Clear Order
2.	Logout
3.	Login Different user
4.	Clear another Order
V	Are the changes being processed correctly, and does the new login show a clean order? YES
R	

V = Validation Point;R = Remarks

Ordering

Add Custom Pizza

Steps	Actions
1.	Select Ingredient
2.	Check if Photo is updated, check if Price changes correctly
3.	Select another Ingredient
4.	Type a name
5.	Select another Ingredient
6.	Remove one of the previously selected ingredients
7.	Press Add to Order
8.	Check if the ingredients are listed correctly in the order
9.	Select another Ingredient
10.	Press Add to Order
11.	Check if the new pizza was added correctly
V	Are the changes being processed correctly, and does the new record show the? YES
R	

V = Validation Point;R = Remarks

Add Non-pizza

Steps	Actions
1.	Click on Side Order
2.	Click on the shopping cart next to a non-pizza product
3.	Click on the shopping cart next to another non-pizza product
4.	Click on the shopping cart next to the in step 2 chosen non-pizza product
5.	Check if the first product has amount "2" and the other product amount "1"
V	Are the changes being processed correctly, and does the new record show the different user as employee who changed it? YES
R	

V = Validation Point;R = Remarks

Current Order

Steps	Actions
1.	Select Ingredient
2.	Press Add to Order
3.	Press on the – under the pizza to remove it from the order list
4.	Select Ingredient
5.	Press Add to Order
6.	Press on the + under the pizza to increase the amount by 1
7.	Press on the + under the pizza to increase the amount by 1
8.	Check if Price and Total Price are updated correctly
9.	Press on the - under the pizza to decrease the amount by 1
10.	Click on the red X to remove the pizza from the order list
V	Are the changes being processed correctly, and does the new record show the different user as employee who changed it? YES
R	

V = Validation Point;R = Remarks

Delivery Information

Steps	Actions
1.	Click on Delivery Info
2.	Type in Customer Information and pick a different delivery date and time
3.	Select another tab
4.	Go back to Delivery Info
5.	Check if data remains the same
V	Are the changes being processed correctly, and does the new record correct delivery date and other info? YES
R	

V = Validation Point;R = Remarks

Confirm Order

Steps	Actions
1.	Click on Confirm Order
2.	Check the error message
3.	Select an ingredient
4.	Add to Order
5.	Click on Confirm Order
6.	Check if Delivery Info is shown
7.	Fill in Delivery Info
8.	Click on Confirm Order
9.	See if confirmation page is shown
10.	Click on Cancel
11.	Click on Confirm Order
12.	Click on OK
V	Are the changes being processed correctly, and does the new record show the order that is confirmed? YES
R	

V = Validation Point; R = Remarks

Chapter 5: Conclusion

After reviewing our work, the conclusion is that after many adjustments the system works. As good as it is now, there can still be made many adjustments/improvements. However in the time was given that three persons can work on this project, the overall results are satisfactory in our opinion. The report covers the entire course of the project and results are there were needed. This conclusion chapter is more a description of the process rather than a summary to give all results and facts.

The first weeks the work progressed slower than expected, in the last weeks the pace was increased to finish on time. At several moments, the progress and pace was commented on that "the progress was not good". The results of this report should be sufficient evidence that work was done properly.

In chapter 1 the problem description is given, the main goal is to give a clear understanding what we are doing. Together with the given problems. The approach in chapter 2 gives the explanation how we are going to work with the problems that arised in chapter 1 and future planning. Chapter 3 is about the results that derived from chapter 1 and 2 problems and issues. In chapter 3 the system "how we make it" is discussed and specified. Various specifications of the system but also the users are specified. In chapter 4 we evaluate the system, this is done with testing templates and making people work with the system other than us. We did not perform interviews or step by step evaluation with the people other than us, we just showed the system and asked for feedback. Basically we treated people who we asked to give feedback as customers, questions like what do u think if the design and such. The final chapter 5 is our conclusion, as said before the conclusion in this report is not a summary of results and facts. In the report all aspects are given clear explanation, hence the conclusion would just be a simple summary of our previous reports. The conclusion as it is now hopefully gives a understanding in the steps taken to make the report.

Appendix A:

User manual

Table of contents:

1. Order placement
 - a. Customizing pizzas
 - i. Choosing ingredients
 - ii. Removing ingredients
 - iii. Changing amount/Deleting of a pizza
 - iv. Adding another pizza
 - b. Picking Non Pizza products
 - i. Viewing a Non Pizza product description
 - ii. Adding a Non Pizza product
 - iii. Changing amount/Deleting of a Non Pizza product
 - c. Checkout
 - i. Viewing an order
 - ii. Entering delivery information
 - iii. Confirming an order
2. Order processing
 - a. Viewing Orders
 - b. Accepting Orders
 - c. Rejecting Orders
 - d. Completing Orders
3. Administration
 - a. Ingredients
 - i. Viewing ingredient list
 - ii. Adding a new ingredient
 - iii. Modifying an ingredient
 - iv. Deleting an ingredient
 - b. Non Pizza
 - i. Viewing Non Pizza list
 - ii. Adding a new Non Pizza
 - iii. Modifying a Non Pizza
 - iv. Deleting a Non Pizza
 - c. Employees
 - i. Adding an employee
 - ii. Removing an employee

1. Order placement

Order placement involves three steps: customizing pizzas, picking non pizza products and entering customer information. These pages are accessible by clicking on the corresponding tab.

a. Customizing pizzas

To customize pizzas click on the Pizza tab and use the following functions:

i. Choosing ingredients

Under "Choose your ingredient", click on the name of the ingredient

ii. Removing ingredients

Under "Current Pizza Creation", click on the red cross next to the ingredient you want to delete

iii. Changing amount of a pizza

Under "Pizza Creations", click on the + or – button of a pizza increase or decrease the amount

iv. Deleting a pizza

Under "Pizza Creations", click on the red cross of a pizza

v. Adding another pizza

Under "Current Pizza Creation", click on the "Create new pizza" button

b. Picking Non Pizza products

To pick Non Pizza product click on the Non Pizza tab and use the following functions:

i. Viewing a Non Pizza product description

Under "Choose your product", click on the name of the Non Pizza

ii. Adding a Non Pizza product

When viewing a Non Pizza product, click on "Add to order"

iii. Changing amount of a Non Pizza product

Under "Chosen Products", click on the + or – button of a Non Pizza increase or decrease the amount

iv. Deleting a Non Pizza product

Under "Chosen Products", click on the red cross of a Non Pizza

c. Checkout

To Checkout click on the Checkout tab and use the following functions:

i. Entering delivery information

Click on "Complete Order" and enter the delivery information

ii. Confirming an order

After entering delivery information, click on "Submit"

2. Order processing

To process orders, employees have to log in to the admin page using their username and password.

a. Viewing Orders

Click on Orders to see a list of Orders

Click on an Order number to see the details of that order

b. Accepting Orders

While viewing an order, under "Change Status Into", click on "In Delivery"

c. Rejecting Orders

While viewing an order, under “Change Status Into”, click on “Cancel”

d. Completing Orders

While viewing an order, under “Change Status Into”, click on “Completed”

3. Administration

To gain access to the following functionalities an Administrator needs to log in to the admin page using his username and password.

a. Ingredients

i. Viewing ingredient list

Under the admin menu, click on “View/Change/Delete Ingredient”

ii. Adding a new ingredient

Under the admin menu, click on “New Ingredient”

Fill in the information, pick an image, and press “Submit”

iii. Modifying an ingredient

Under the admin menu, click on “View/Change/Delete Ingredient”

Click on the pencil of the ingredient you want to modify

Change the information and press “Submit”

iv. Deleting an ingredient

Under the admin menu, click on “View/Change/Delete Ingredient”

Click on the red cross of the ingredient you want to delete

b. Non Pizza

i. Viewing Non Pizza list

Under the admin menu, click on “View/Change/Delete Non-Pizza”

ii. Adding a new Non Pizza

Under the admin menu, click on “New Non-Pizza”

Fill in the information, pick an image, and press “Submit”

iii. Modifying a Non Pizza

Under the admin menu, click on “View/Change/Delete Non-Pizza”

Click on the pencil of the Non-Pizza you want to modify

Change the information and press “Submit”

iv. Deleting a Non Pizza

Under the admin menu, click on “View/Change/Delete Non-Pizza”

Click on the red cross of the Non-Pizza you want to delete

c. Employees

i. Adding an employee

Under the admin menu, click on “Employees”

Click on “New Employee”, fill in username and password, press “Submit”

ii. Removing an employee

Under the admin menu, click on “Employees”

Click on the red cross of the Employee you want to delete