UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Web Applications, A.Y. 2019/2020
Master Degree in Computer Engineering
Master Degree in ICT for Internet and Multimedia

Homework 1 – Client-side Design and Development

Submission date: 24 April 2020

Last Name	First Name	Student Number
Favaro	Simone	1234367
Foscarin	Daniele	1242074
Molon	Alberto	1237327
Neri	Michael	1232224
Shenaj	Donald	1238939

Objectives

The purpose of our project is to build a takeaway pizzeria web application, serving the management of the orders and the deliveries, while customers can spoil themself with a user-friendly interface to place their orders in a fully automated way, create new pizzas and easily pay their meals.

This system lightens employees workload by eliminating the needs for a specific person who takes ordinations and the cash payments. Moreover, our application offers some features for the optimization of the pizza making workflow, like an efficient queue management, regarding ovens and delivery guys availability.

On the other hand, customers can monitor their order, witnessing the order status that is continuously updated by the staff. Also, to increase fidelity, the webapp offers a single order discount for users which create the most trendy pizzas (that are the most requested custom made pizza of the week).

Ripperoni Pizza web application offers to customers the same service provided by a normal takeaway pizzeria, without the need of employees apart from cooks and delivery guys; while caring especially about quality of user experience and speed of delivery. All that makes Ripperoni Pizza the perfect tradeoff between high standards but huge management costs of normal takeaway pizzerias and the optimization but carelessness of delivery companies.

Main functionalities

To summarize, the main functionalities for customers are:

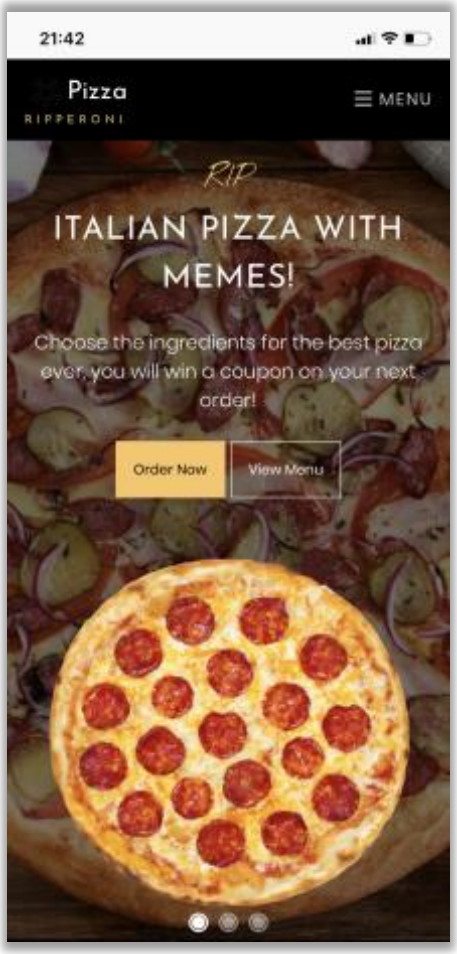
- create new pizzas from scratch;

- order both custom made and traditional pizzas;
- pick a delivery time that goes well with the order queue;
- get a secure online payment method;
- monitor the delivery status;
- get order discount based on the popularity of the pizza they create;
- display of the full order history and the list of created pizzas.

on the other hand, main functionalities on the employees side are:

- smart order queue management for both cooks and delivery guys;
- automatic smart routing and display of the next address to deliver for delivery guys;
- manual update of the order status for both cooks and delivery guys;
- display of the oven capacity and the number of pizzas ready to be prepared.

Design choices



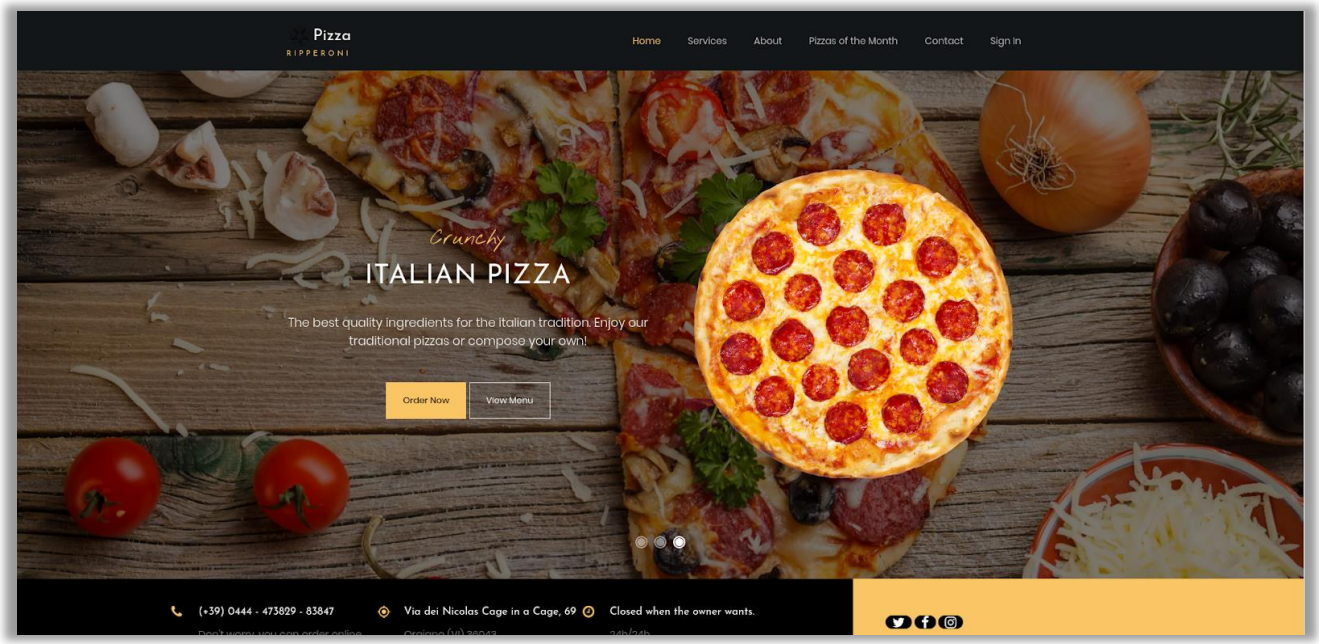
We have decided to develop our website using Bootstrap CSS Framework which is a developing tool that helps to create a nice and responsive view. For this reason, our website is able to adapt even in smaller screens; the upper navbar is always present and the main content of the pages is adjusted responsively depending on the screen size.

Concerning the general theme of the p ages, we also gave particular attention to the colors themes, fonts styles and button animations, to keep consistency across the web site. For the same consistency reasons, we used only Flaticon icons, available at https://www.flaticon.com/, Icomoon font, available at https://icomoon.io/ and Open Iconic, available at https://useiconic.com/open.

Even if we use a premade Colorlib template available at https://colorlib.com/, it was just for the sake of outlining a consistent guideline throughout the whole web site. We fully edit it in order to obtain complete freedom on the design. The chunks of code we decide to take are the following:

- animation management, created by Daniel Eden and avaiable at https://daneden.github.io/animate.css/ under MIT License http://opensource.org/licenses/MIT/;
- slides management, created by David Deutsch and avaiable at https://owlcarousel2.github.io/OwlCarousel2/.

Homepage



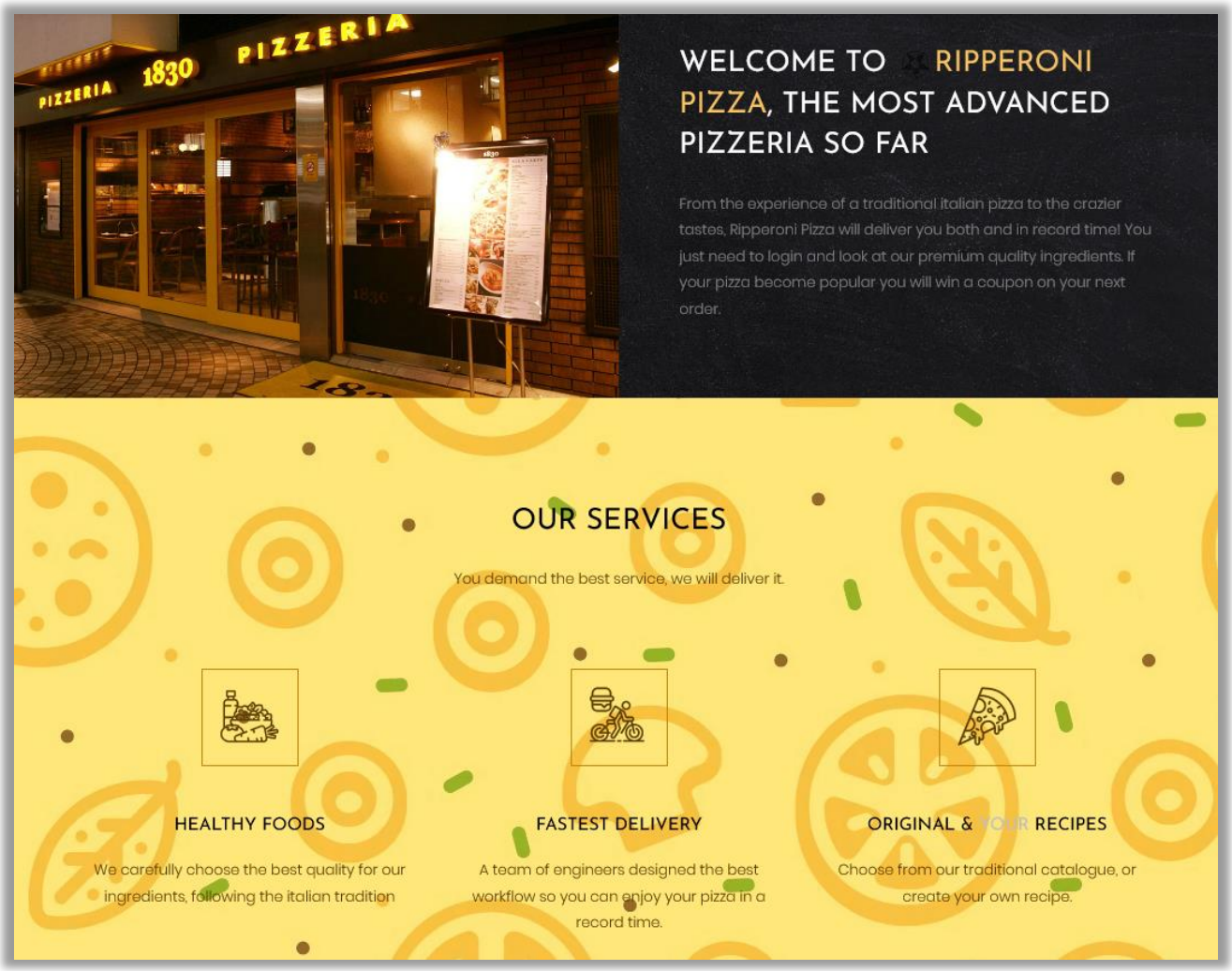
The homepage is the first view of our website, where you can find all the main features of the webapp. It is composed by: a navbar on the top, a central body and a footer on the bottom.
In the navbar, the user can find: the pizzeria logo on the left and some links on the right. The navbar is present in all the html pages of the webapp. Furthermore, it is set to be fixed on the top and disappears while scrolling the webpage.
The links on the right are: Home, Services, About, Contact and Sign-in:

- 'Home' (and also the logo) acts as link that take the user back to the homepage;
- 'Services' is connected to the relative fragment of the homepage which describes the services offered by the webapp;
- 'About' is a link that leads to another html page where the Ripperoni team is described;
- 'Contact' is connected to the relative fragment of the homepage where there is a form used to contact the Ripperoni team;
- 'Sign-in' opens a pop-up that allows the user to log-in or sign-up into the webapp.

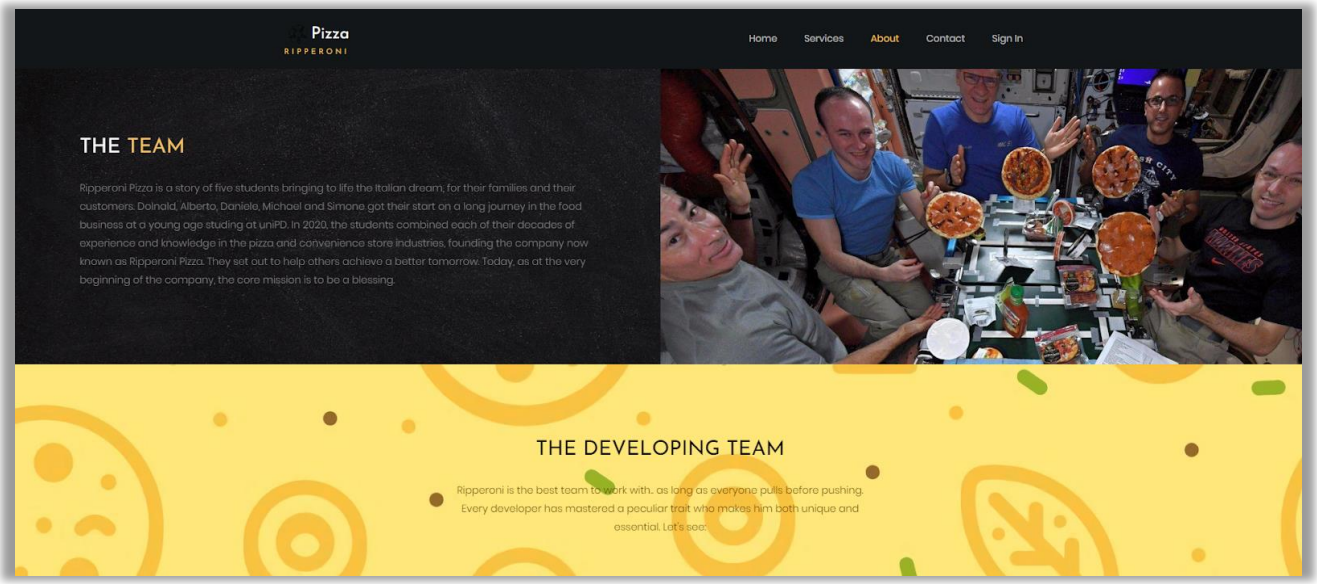
At the top of the page there is a slider that allows the user to see the pizza menu and make an order. It is visible in the foreground so that the user can immediately read the menu (and possibly see the new pizzas added) and/or order quickly (after doing the log-in).
At the bottom of the page you can also find the social contacts.

Concerning the design, the color scheme used in the homepage, as well as the other pages, is based on dark grey and yellow colors. The use of a dark gray shade for the backgrounds aims to recall the aspect of a chalkboard commonly used in pubs and other food related places, while maintaining an elegant feel. The yellow elements offer a contrast to the dark backgrounds and they recall the aspect of a paper menu or flyers.

Moreover, the webapp is designed to offer a fun social network-like experience to the customer, so we have introduced playful and cartoonish elements, like the interactive icons and the endearing descriptions.


PIZZERIA 1830 PIZZERIA
WELCOME TO RIPPERONI PIZZA, THE MOST ADVANCED PIZZERIA SO FAR
From the experience of a traditional italian pizza to the crazier tastes, Ripperoni Pizza will deliver you both and in record time! You just need to login and look at our premium quality ingredients. If your pizza become popular you will win a coupon on your next order.
OUR SERVICES
You demand the best service, we will deliver it.
HEALTHY FOODS
We carefully choose the best quality for our ingredients, following the italian tradition
FASTEST DELIVERY
A team of engineers designed the best workflow so you can enjoy your pizza in a record time.
ORIGINAL & YOUR RECIPES
Choose from our traditional catalogue, or create your own recipe.

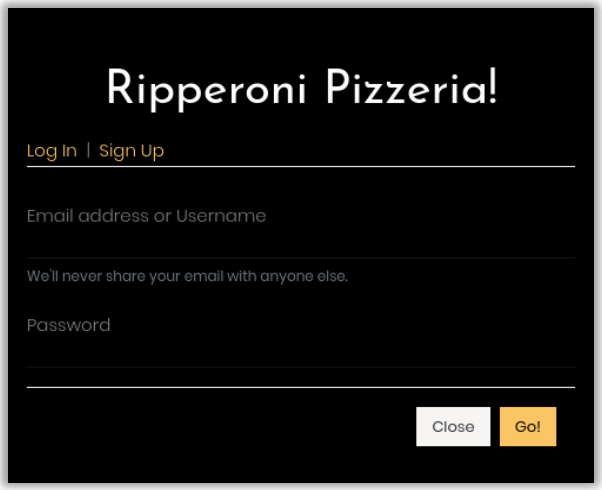
About Us


Pizza
RIPPERONI
Home
Services
About
Contact
Sign In
THE TEAM
Ripperoni Pizza is a story of five students bringing to life the Italian dream, for their families and their customers. Donald, Alberto, Daniela, Michael and Simone got their start on a long journey in the food business at a young age studing at uniPD. In 2020, the students combined each of their decades of experience and knowledge in the pizza and convenience store industries, founding the company now known as Ripperoni Pizza. They set out to help others achieve a better tomorrow. Today, as at the very beginning of the company, the core mission is to be a blessing.
THE DEVELOPING TEAM
Ripperoni is the best team to work with, as long as everyone pulls before pushing. Every developer has mastered a peculiar trait who makes him both unique and essential. Let's see:

In this page, the user can read the funny description of the Ripperoni professional team, from the origins of a simple dream to its consecration.
At the bottom of the page, the user can also find a form to contact the team for any needs, advice and hiring.

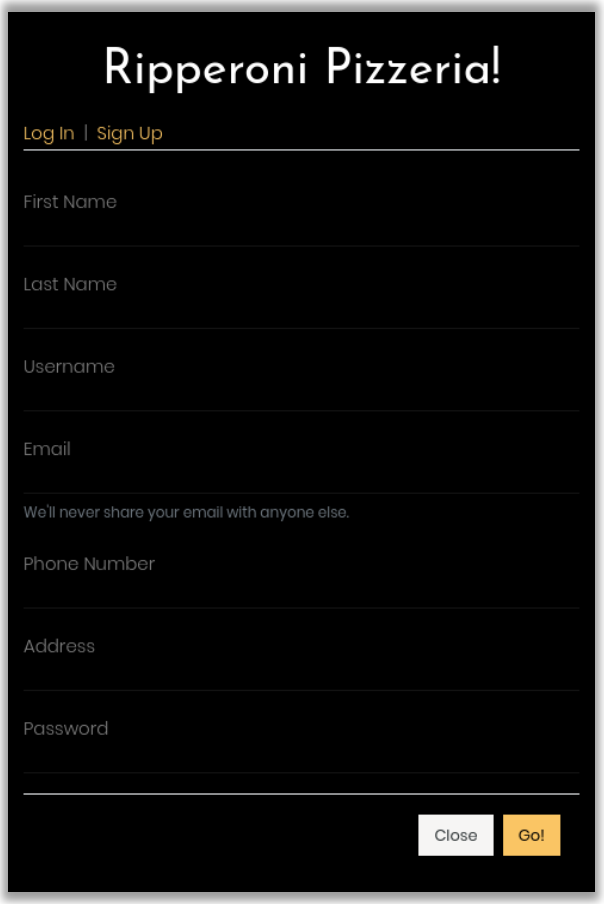
Login

After clicking on the sign-in link on the navbar a form pops up, that is used both for registration and login of customers and employees in order to access the services provided by the respective dashboards.
When logging in, a user is asked to insert his account credentials, which are username or email and password. Here we used form validation in order to check if the information submitted by the user are plausible, indeed if there is not text, it appears a message saying to insert a valid username or email.
After finishing enter the account credentials, the user has to click the 'Go!' button to access the actual dashboard. The dummy credentials we used in the



mock up are:

- Username:NicolasCage or NicolasKimCoppola@yahoo.ue and password:ripperoni for the customer dashboard;
- Useraname:NicolasCageCook and password:ripperoni for the cook dashboard;
- Username:NicolasCageDel and password:ripperoni for the delivery guy dashboard.



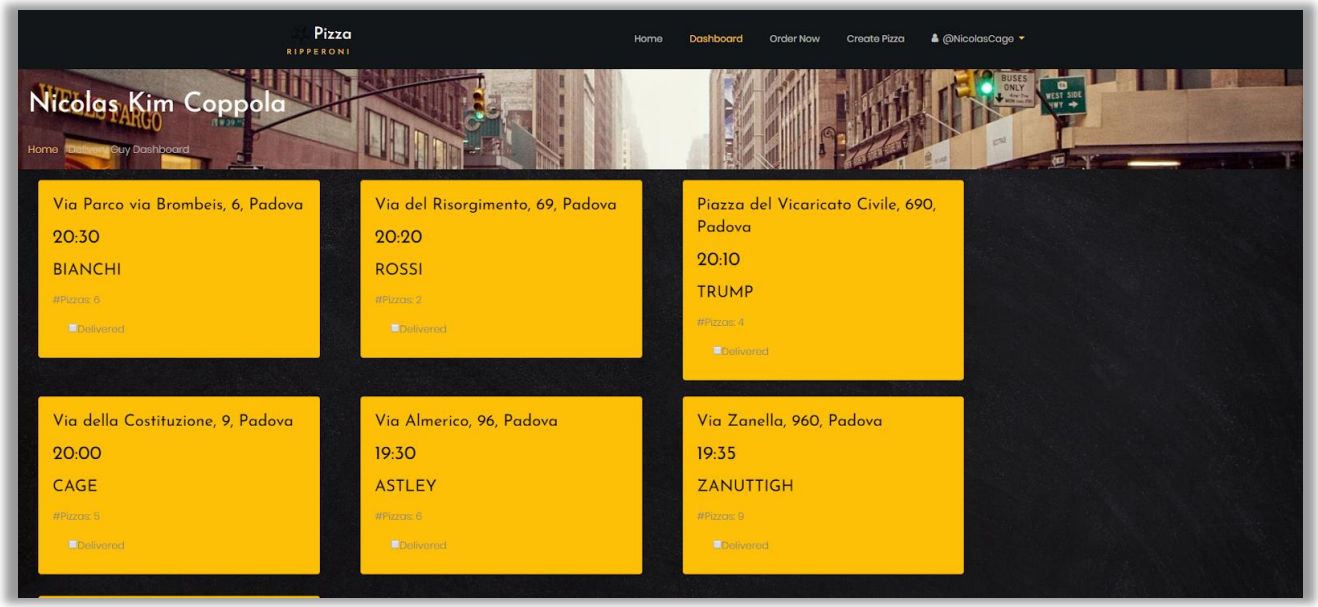
Sign Up

The sign-up part of the form, is used by new user who has to register their credential in the application for the first time.
When signing up, a user is asked to insert the information required to be uniquely identified (username, email and password) and some additional information (first name, last name) which are used to garnish the dashboards. Phone number and delivery address are also required, because they are extremely useful in order to provide the necessary information for the delivery of the orders.

Even in this section is used form validation in order to check the information submitted by the user. If the form is not complete when clicking the 'Go!' button, the application notify it with messages in the respective non filled fields. Also if the user tries to put numbers in the First and Last name fields, the application notifies it. Finally the same thing happens when a string is submitted in the phone contact field.

Dashboards

After the login the user enters his respective dashboard. There are three different dashboards for cooks, delivery guys and customers.
For each one of them we use json files in order to simulate the server side interactions. They contain dummy information that are extracted using get json requests.



The delivery guy dashboard is the main service used by delivery guys to do their work. This page contains the information used by the employees to do their job without being distracted by unnecessary graphic contents. It contains a grid of cards where are listed all the deliveries that has to be done. For each one of them, the information specified are:

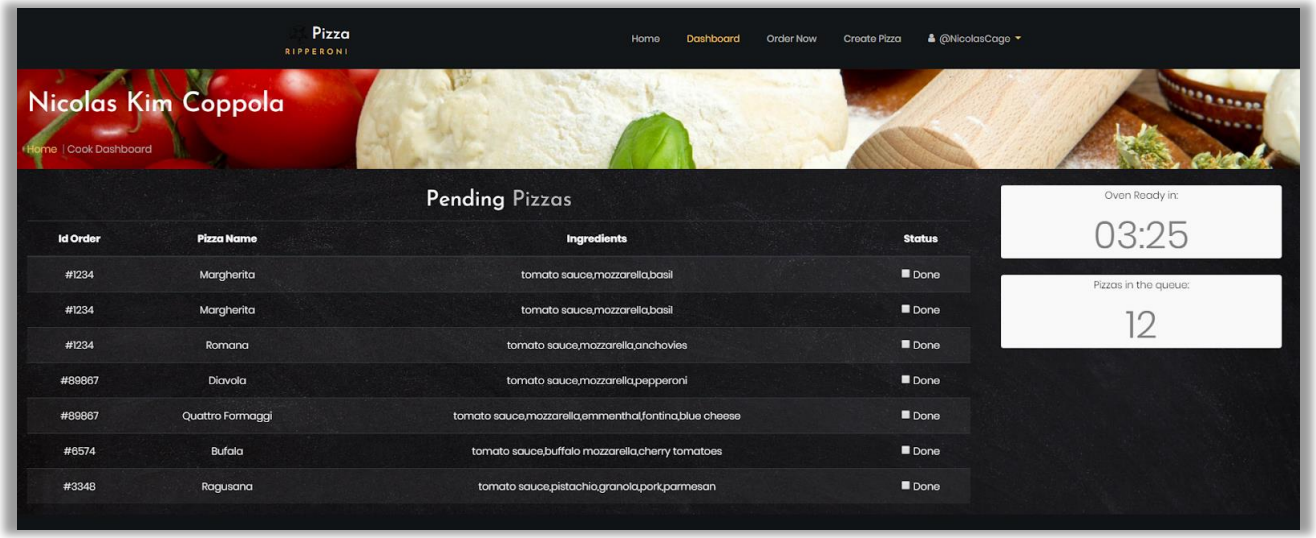
- delivery address;
- delivery time decided by the customer;
- surname of the customer;
- number of pizza contained in the order;
- a checkbox that has to be ticked when the order is successfully delivered: this information is used to modify the order status from baking to delivering. Furthermore, when a delivery is ticked, it fades away.

The cards that are placed in the same horizontal row contain the deliveries that should be done in the same trip, so before the next row the delivery

guy needs to go back to the pizzeria. The total number of orders assigned to the same trip will be controlled by the backend.

The following is a chunk of json data used to mimic the server interaction. It contains a dummy list of orders to be delivered.


```
[
    {
        "IDOrder": "#1234",
        "Address": "Via Parco via Brombeis, 6, Padova",
        "Surname": "Bianchi",
        "DeliveryTime": "20:30",
        "Pizzas": 6
    },
    {
        "IDOrder": "#1234",
        "Address": "Via del Risorgimento, 69, Padova",
        "Surname": "Rossi",
        "DeliveryTime": "20:20",
        "Pizzas": 2
    },
```

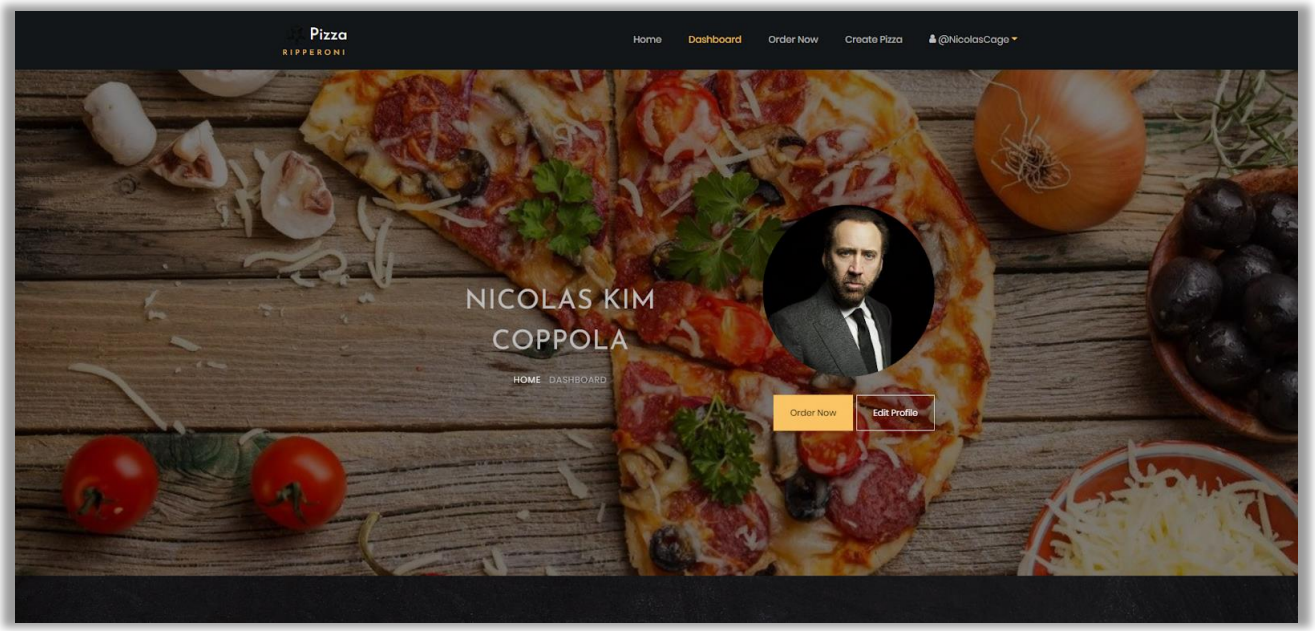



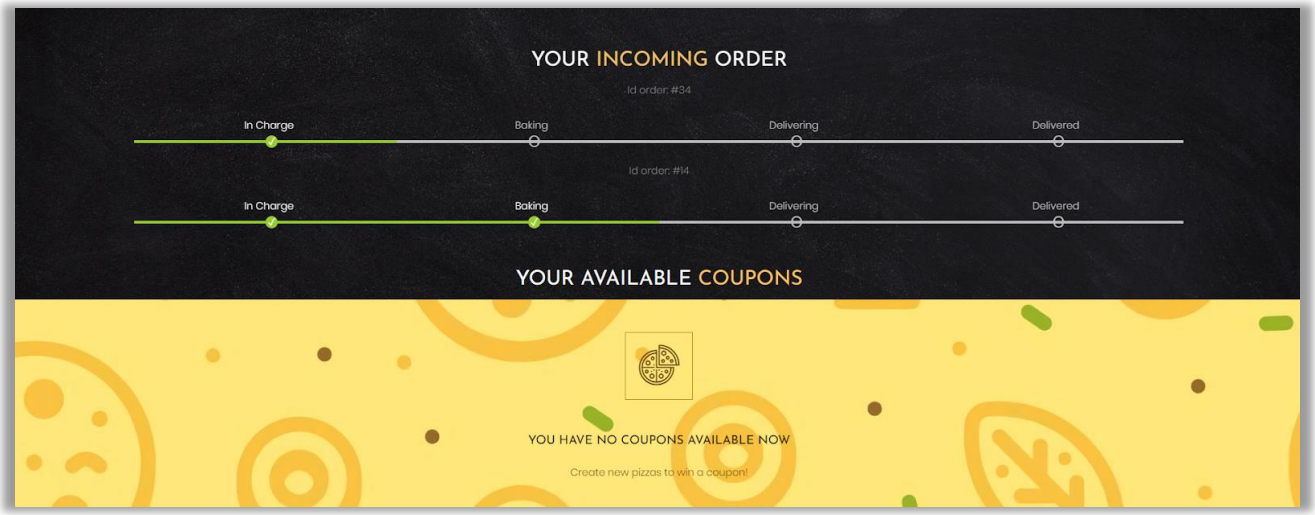
The cook dashboard is the service used by cooks to monitor the order queue. It contains:

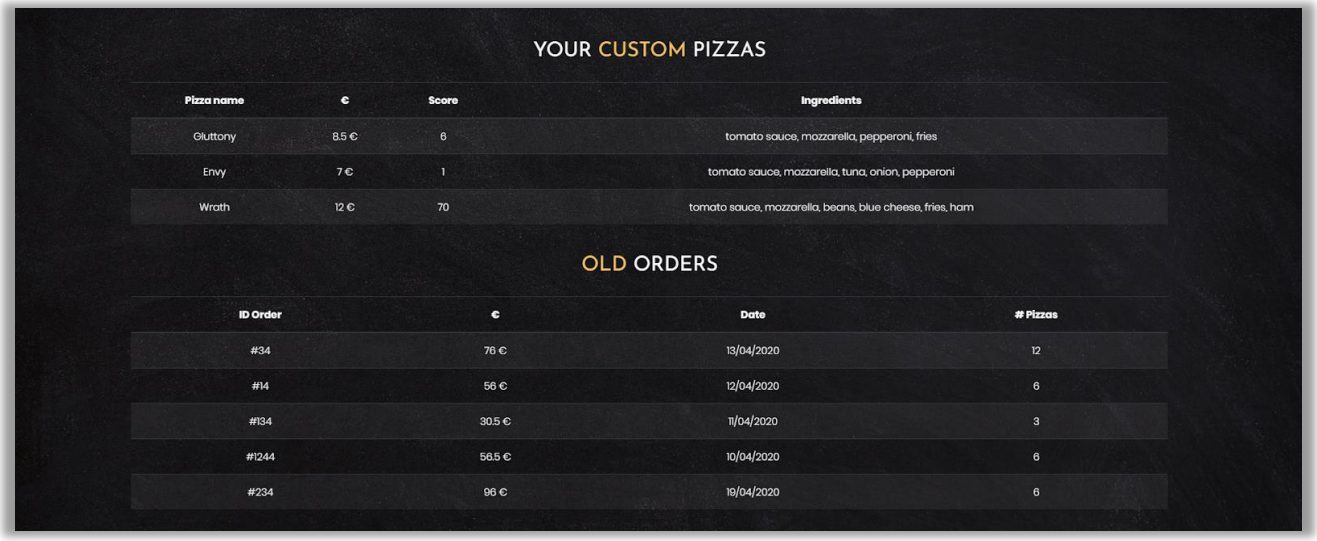
- all the pending pizzas, which has to be baked. For each one of them is included a ‘Done’ flag which updates the order status from in charge to baking and eliminates the pending pizza;
- the number of pizza in queue requested by customers;
- the oven countdown which represents both the cooking time of the pizzas inside the oven and its availability for the next batch of pizzas to be baked.

The following is an chunk of json data used to mimic the server interaction. It contains a dummy list of pending pizza to be baked:


```
[
    {
        "IDOrder": "#1234",
        "PizzaName": "Margherita",
        "ingredients": ["tomato sauce","mozzarella","basil"]
    },
    {
        "IDOrder": "#1234",
        "PizzaName": "Margherita",
        "ingredients": ["tomato sauce","mozzarella","basil"]
    },
```


Pizza
RIPPERONI
Home
Dashboard
Order Now
Create Pizza
@NicolasCage
NICOLAS KIM
COPPOLA
HOME DASHBOARD
Order Now
Edit Profile

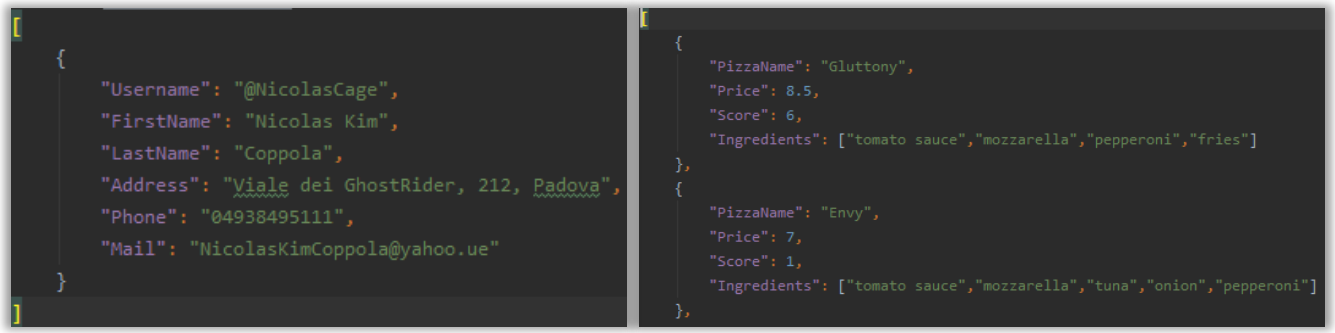

YOUR INCOMING ORDER
Id order: #34
In Charge
Baking
Delivering
Delivered
Id order: #14
In Charge
Baking
Delivering
Delivered
YOUR AVAILABLE COUPONS
YOU HAVE NO COUPONS AVAILABLE NOW
Create new pizzas to win a coupon!


YOUR CUSTOM PIZZAS
Pizza name
€
Score
Ingredients
Gluttony
8.5 €
6
tomato sauce, mozzarella, pepperoni, fries
Envy
7 €
1
tomato sauce, mozzarella, tuna, onion, pepperoni
Wrath
12 €
70
tomato sauce, mozzarella, beans, blue cheese, fries, ham
OLD ORDERS
ID Order
€
Date
Pizzas
#34
76 €
13/04/2020
12
#14
56 €
12/04/2020
6
#134
30.5 €
11/04/2020
3
#1244
56.5 €
10/04/2020
6
#234
96 €
19/04/2020
6

The customer dashboard is the service used by customers to order pizzas, create new ones, see the order status and the complete history of the orders. It contains:

- an overview of the profile (profile pic, fist and last name);
- buttons which links to the Order Now and Edit Profile pages;
- coupon list owned by the customer;
- the status of incoming orders (identified by the order number), displayed with a bar which highlights the main progression stages of the order (in charge, baking, delivering, delivered);
- the list of custom pizza created by the customer;
- the score of the created pizza;
- the history of the previous orders.

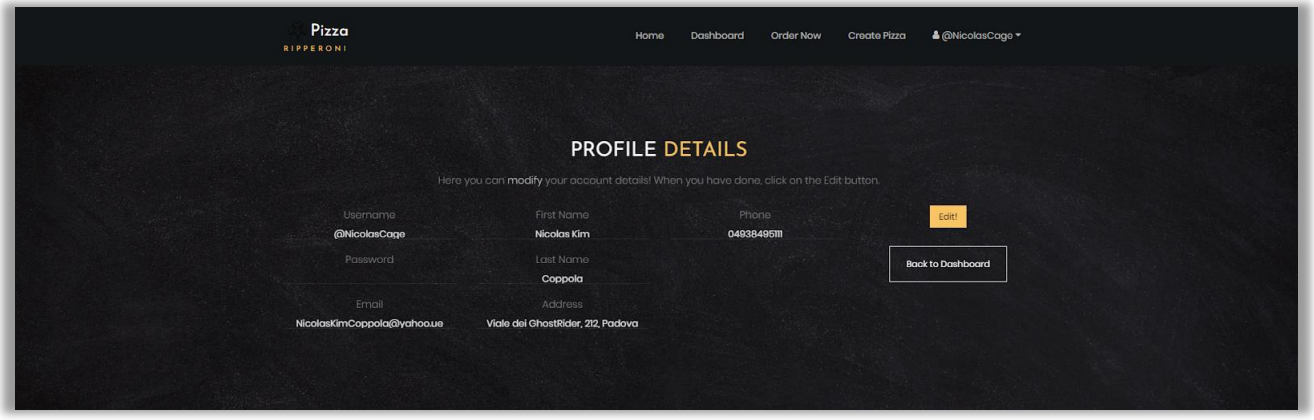
The following are chunks of json data used to mimic the server interaction. They contain a dummy account information and dummy lists for past created pizzas.



```
[
    {
        "Username": "@NicolasCage",
        "FirstName": "Nicolas Kim",
        "LastName": "Coppola",
        "Address": "Viale dei GhostRider, 212, Padova",
        "Phone": "04938495111",
        "Mail": "NicolasKimCoppola@yahoo.ue"
    }
]
```

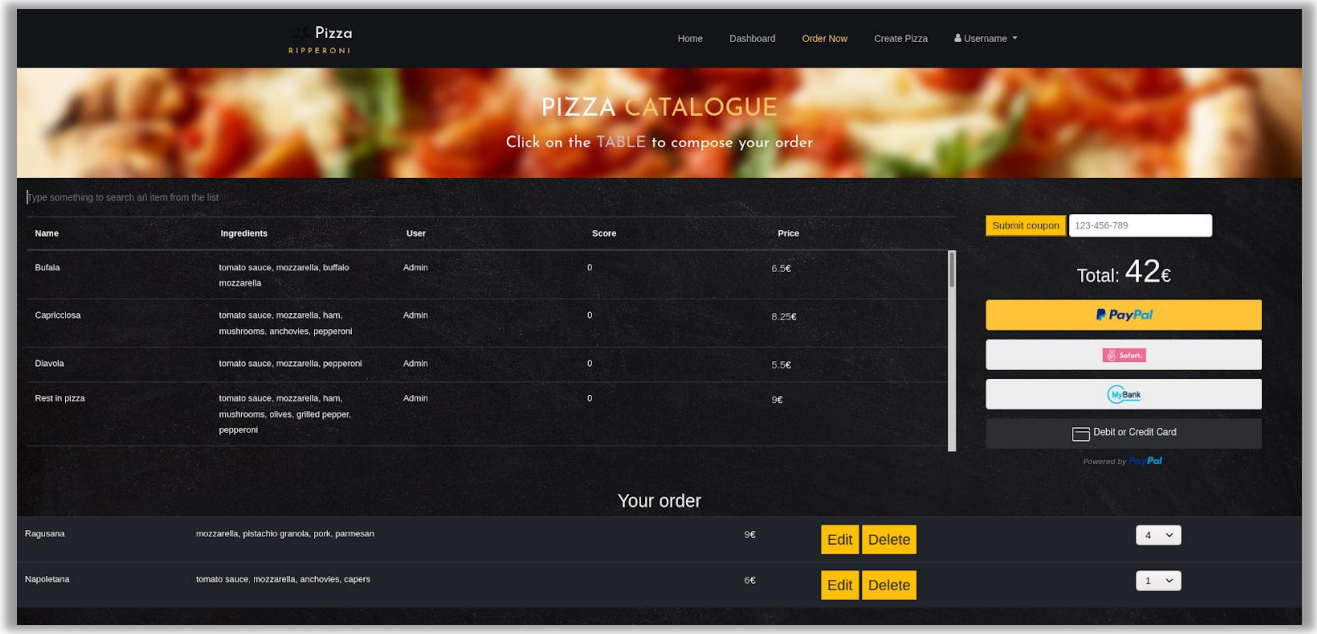
```
[
    {
        "PizzaName": "Gluttony",
        "Price": 8.5,
        "Score": 6,
        "Ingredients": ["tomato sauce","mozzarella","pepperoni","fries"]
    },
    {
        "PizzaName": "Envy",
        "Price": 7,
        "Score": 1,
        "Ingredients": ["tomato sauce","mozzarella","tuna","onion","pepperoni"]
    },
```

Edit Profile



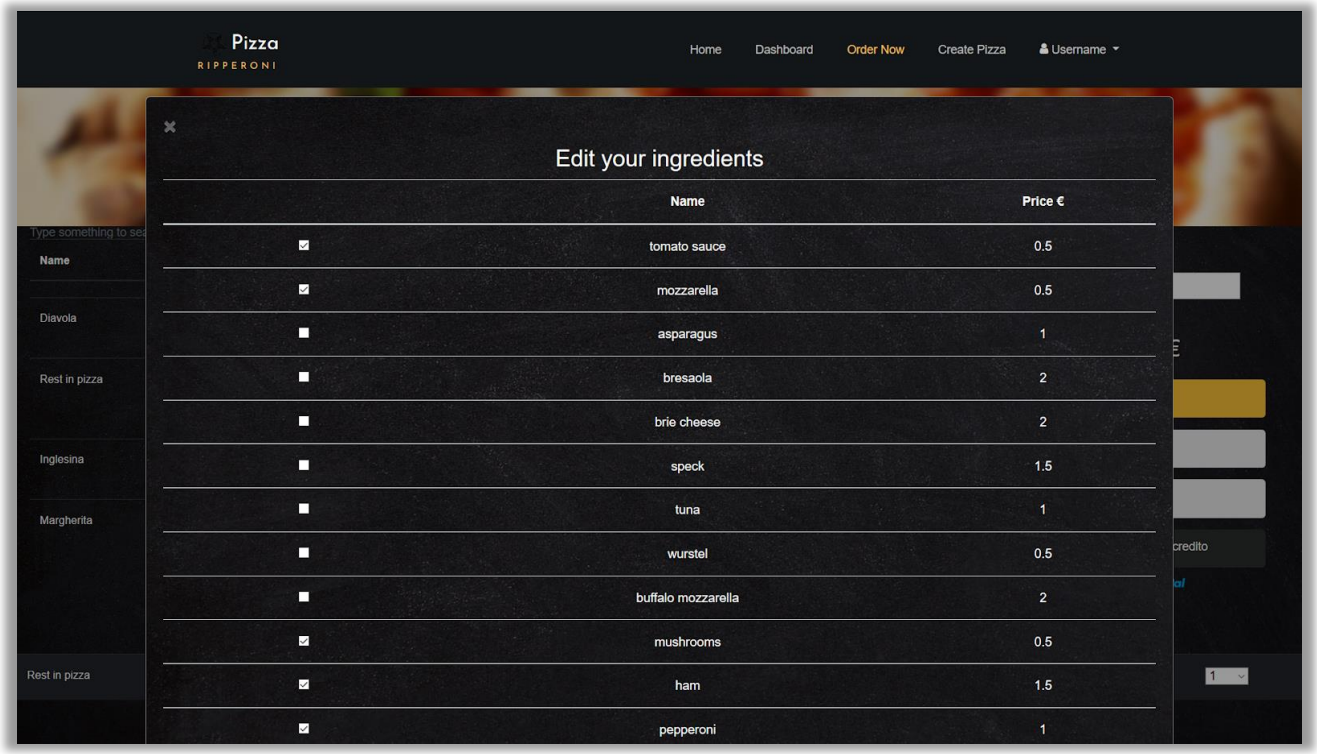
In this page the user can modify the very same information he provided during the registration. The “Edit!” button is used for the actual edit of the fields (the form validation is the same of the sign up form) and the ‘Back to Dashboard’ button redirect user to his dashboard, concluding the editing.

Order



This page contains the pizza catalogue from which the user can choose the pizzas he wants to order and there are implemented many functionalities. The main table contains all the traditional pizzas and the pizzas created by other customers. In the columns there are information about the name, the ingredients, the user who created the pizza, the success score, and the price. Notice that the traditional pizzas appear created by the “Admin”, while the custom one will have the name of the user.

The customer can look for a pizza just by typing its name or ingredients in the search list on the top of the table, or by simply scrolling the table. The webpage is meant to be very responsive so the user just needs to click on a row (in any point) to add the selected pizza to the “your order” table. If the user clicks the same row more than once, the order list will contain the same pizza the number of times the row has been clicked. After selecting a pizza, the user can edit it by pressing the edit button. At this point, a popup will open and the user can choose the ingredients to add (or remove) to (or from) the chosen pizza. Notice that he could do multiple choices by adding or removing an undefined number of ingredients.



The price of an edited pizza will be modified according to the price of the ingredients that have been added or removed after the submit button is pressed. Simultaneously the displayed ingredients in the your order table will include the new selected ingredients for the edited pizza, and the name of that pizza will change to “Edited”, representing that is not a standard pizza. At this point the price is all computed in the javascript but later on will be properly verified by the backend to avoid cheats. In the same table the user can delete a pizza from his order list or modify the quantity distinct pizzas using a dropdown menu. We have emulated the retrieving of data operations using ajax calls in the order.js file which take the ingredients data from the ingredients.json and pizza catalogue data from complete-pizza-catalogue.json.

When the user has composed his order, he will be able to see the total amount on the right of the page (which is updated at each change of the order table) and he can proceed to the payment by selecting the payment method.

Over to the payment section there is a button and a text field that allows the user to enter a coupon code, which will be verified with a form validation that check if the code follows the correct format.
The following are chunks of json data used to mimic the server interaction. They contain the lists of pizzas and pizza ingredients used by the customer to order pizzas.

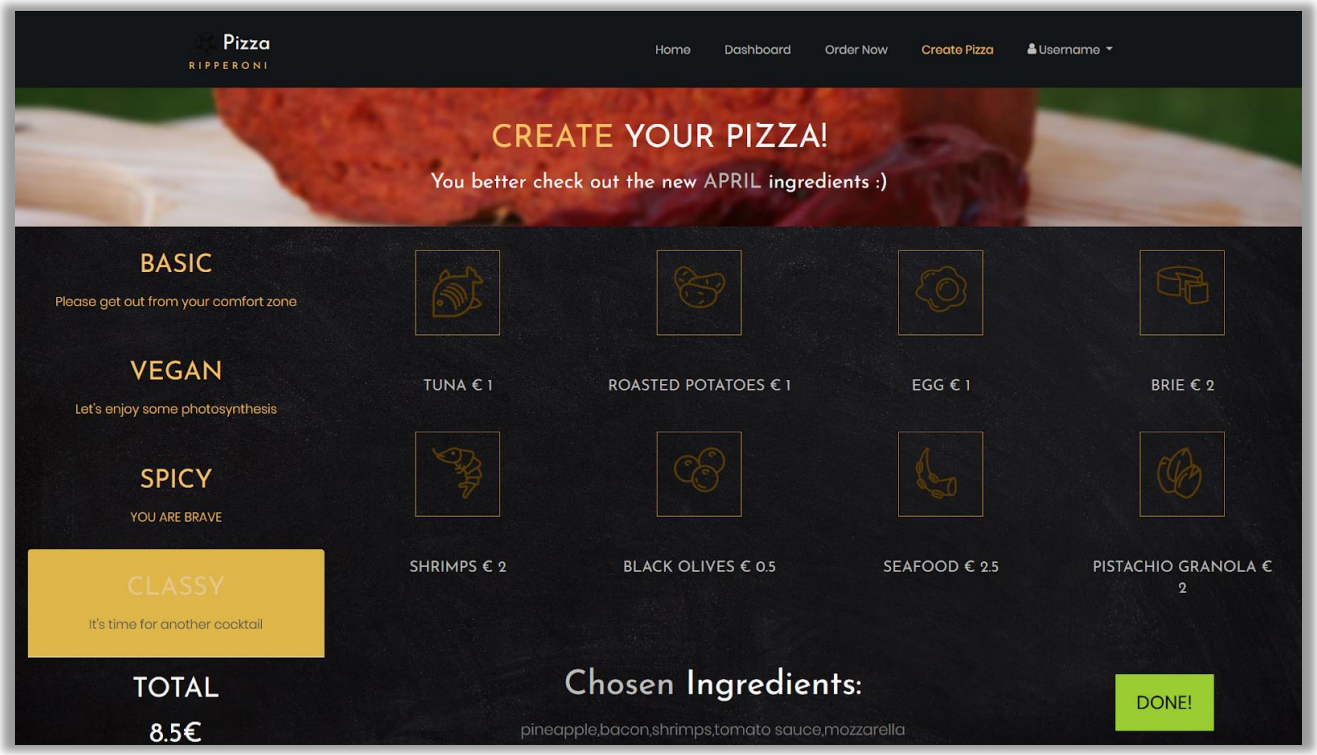

```
[
    {
        "pizza_name": "Bufala",
        "price": "6.5",
        "username": "Admin",
        "score": "0",
        "ingredients": "tomato sauce,mozzarella,buffalo mozzarella"
    },
    {
        "pizza_name": "Capricciosa",
        "price": "6.5",
        "username": "Admin",
        "score": "0",
        "ingredients": "tomato sauce,mozzarella,ham,mushrooms,anchovies,pepperoni"
    },
```



```
[
    {
        "ingredient": "tomato sauce",
        "additional_cost": "0.5"
    },
    {
        "ingredient": "mozzarella",
        "additional_cost": "0.5"
    },
```

Create Pizza

In this page, the user can create his favorite pizza! To do this, he can choose from the pool of ingredients which is divided in four categories: basic, vegan, spicy and classy.
The user see the total price of the pizza that is updated every time that a new ingredient is inserted (clicking on it), which is also visible in the chosen ingredients section.



Starting from a base price, it increases/decreases when an ingredient is added/removed.
At the end of the creation phase, the user has to press the done button, to confirm the pizza and give it a name.

The following is an chunk of json data used to mimic the server interaction. It contains the list of ingredients used by the customer to create the pizzas.
To retrieve the correct icons, in the json there is an “icon” attribute for each ingredient which contains the css class and when added to an html div tag it displays the icon.

```
[
    {
        "icon": "flaticon-tomato",
        "category": "basic",
        "ingredient": "tomato sauce",
        "additional_cost": "0.5"
    },
    {
        "icon": "flaticon-fish",
        "category": "classy",
        "ingredient": "tuna",
        "additional_cost": "1"
    },
```