**jsDonut**

Technical solution description

**Andrey Krupchak**

**30.10.2018**

**Content**

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Introduction……………………………………………………………… | | 3 |
|  | 1.1 | Project Prerequisite…………………………………………...…... | 3 |
|  | 1.2 | Goals……………………………………………………………… | 3 |
| 2 | Technologies and Frameworks…………………………………………... | | 3 |
| 3 | Additional Features……………………………………………………… | | 4 |
| 4 | Database Scheme……………………………………………………….... | | 4 |
| 5 | Description of Tables and their Relations………………………………... | | 5 |
| 6 | Explication and Implementation…………………………………………. | | 6 |
|  | 6.1 | Subject Area………………………………………………………. | 6 |
|  | 6.2 | Peculiarities of the System………………………………………… | 6 |
| 7 | Architecture……………………………………………………………… | | 6 |
| 8 | Modules………………………………………………………………….. | | 7 |
| 9 | User Interface……………………………………………………………. | | 8 |
|  | 9.1 | Filter………………………………………………………………. | 8 |
|  | 9.2 | Templates…………………………………………………………. | 8 |
|  | 9.3 | Used Libraries…………………………………………………….. | 8 |
|  | 9.4 | Mail Page………………………………………………………….. | 8 |
| 10 | Business Logic…………………………………………………………... | | 9 |
| 11 | Entities and DAO………………………………………………………… | | 9 |
|  | 11.1 | Entities…………………………………………………………….. | 9 |
|  | 11.2 | DAO………………………………………………………………. | 9 |
| 12 | Pages……………………………………………………………………... | | 11 |
| 13 | Junit Tests………………………………………………………...……… | | 17 |
| 14 | Ideas and Features……………………………………………………....... | | 18 |

**1 Introdution**

**1.1 Project Prerequisite**

Online stores have become the part of our lives. Many services have moved from offline to online. And selling desserts is not an exception. The demand for this service has led to the growth of companies carrying out such business. Competition has led to an increase in requirements for the quality of services. First of all, customers appreciate the information visibility, user-friendly interface and speed of service delivery.

* 1. **Goals**

The system represents online store of desserts. Its main goals are providing information about all products, opportunities for choosing the most situable ones and for placing an order.

**2 Technologies and Frameworks**

• IntelliJ IDEA 17.2;

• Wildfly 13.0.0;

• MySQL Server 8.0;

• Maven 3.5.4;

• JPA 2.1;

• Spring Framework 5.0.7;

• JUnit 3.8;

• JSP;

• JSF 2.1;

• REST Web Services;

• RichFaces 4.5.17;

• WebSockets;

• AJAX.

**3 Additional Features**

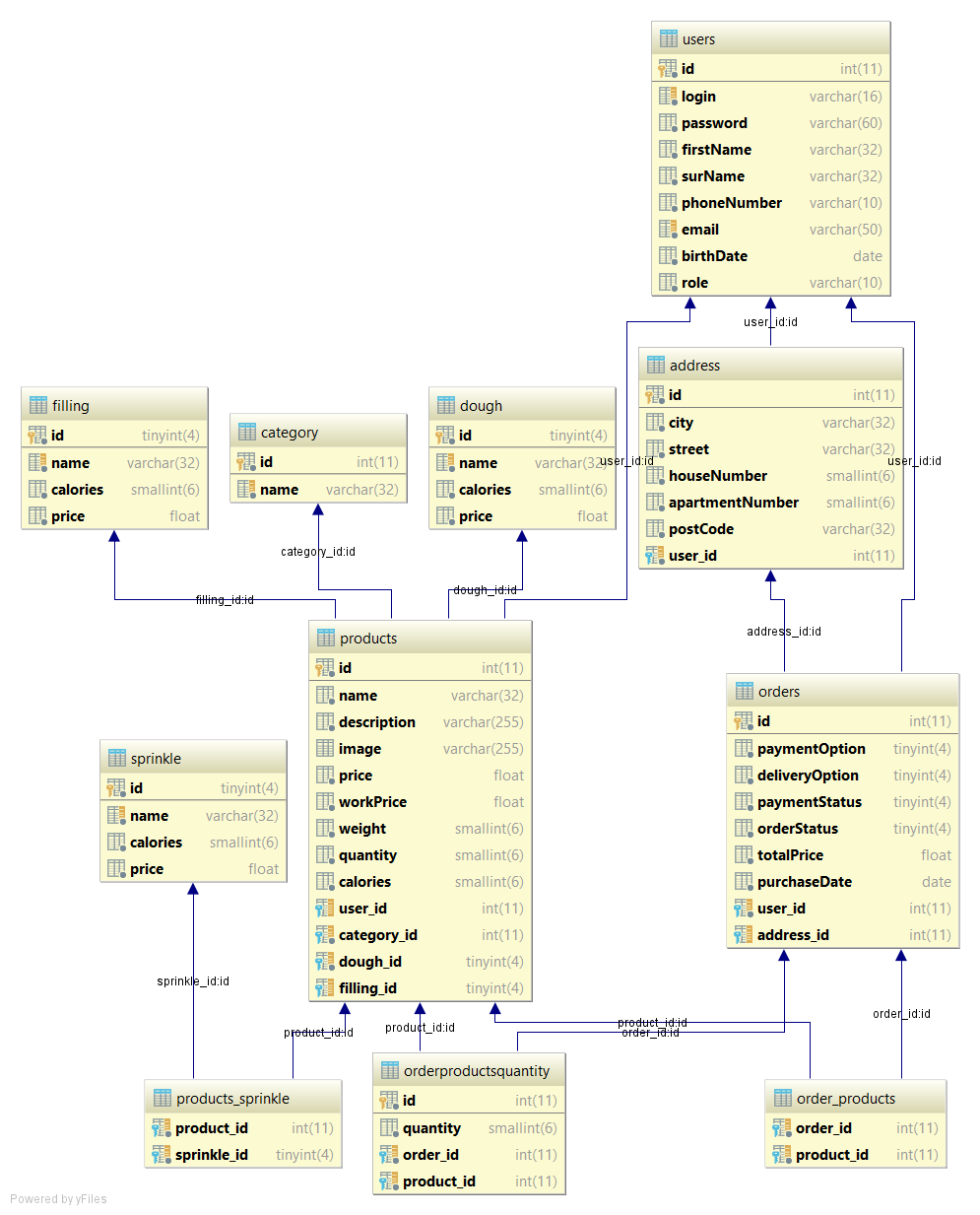
• The ability to select (add / delete) parts that make up the product.

• The ability to add new product categories.

• Designing the cost of the goods, taking into account the composition and weight.

• Information changing in-time when entering new data.

**4 Database Scheme**



**5 Description of Tables and their Relations**

• Table “Users” contains login, password, firstName, surName, birthDate, phoneNumber, email and role.

• Table “Address”: contains city, street, houseNumber, apartmentNumber, postCode and has a link to the table “Users” that allows users have more than one address.

• Table “Orders” contain paymentOption, deliveryOption, paymentStatus, orderStatus, totalPrise, purchaseDate and has links to the table “OrderProductQuantity” (for keeping information about the quantity of goods), to the table “Users” (for keeping information about who placed the order), and to the table “Address” (for keeping information about the details of delivery).

• Table «Order\_Products» keeps the bidirectional relation between the table “Order” and the table “Product”.

• Table “Product” contains name, description, weight, workPrice, price, quantity, calories and has links to the table “Users” (for keeping information about who created this product), to the table “Category” (for keeping information about the type/category of the product), to the table “Filling” (for keeping information about the filling of the product), to the table “Dough” (for keeping information about the dough of the product) and to the table “Sprinkle” (for keeping information about the sprinkle of the product).

• Table “OrderProductQuantity” keeps the unidirectional relation to the table “Order”, to the table “Product” and information about the quantity of the product.

• Table “Category” contains name.

• Table “Dough” contains name, price and calories.

• Table “Filling” contains name, price and calories.

• Table “Sprinkle” contains name, price and calories.

• Table “Products\_Sprinkle” keeps the relation between the table “Product” and the table “Sprinkle”.

**6 Explication and Implementation**

**6.1 Subject Area**

There are the following types of entities:

• Product: Name, Price, Quantity, User.

• Order: Statuses, Products, User.

• User: Name, Addresses.

**6.2 Peculiarities of the System**

The application describes all the subject areas from the task.

The system gives the opportunity to create and edit such entities as “Sprinkle”, “Filling”, “Dough” and “Category of the Product”. “Dough” and “Category of the Product” are obligatory fields for creating a product.

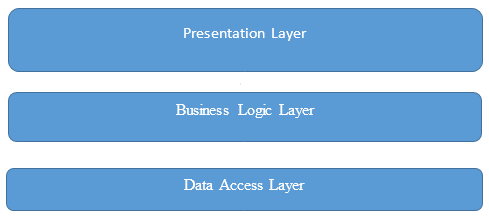
Products contain many parts that allow to combine properties and create the most suitable product according to the description.

Changing the product information leads to the changing information in web-service.

It is necessary that the order contains information about the product(s) and has links to the client and his/her address (depending on the type of delivery). Order statuses created after placing the order. Administrator can edit them according to the execution. For example, checking the actual status of the order payment. After paying the order status changes and administrator starts to complete the order.

**7 Architecture**

Application has a multi-layermonolit architecture.



Presentation layer is represented by Controllers and JSP pages with JavaScript.

Business logic layer consists of the Services and Exceptions.

Data Access Layer is represented by DAO and Entities modules.

**8 Modules**

• Package Controllers contains controllers for GET, POST, DELETE queries from front-end. It is the transmission link between services and front-end. It also contains queries for updating, setting and removing operations using services.

• Package Messages contains classes for sending and receiving messages from other applications using JMS(WildFly).

• Package DTO contains DTO classes. These instances of classes pass between services and controllers.

• Package Services contains classes that describe business logic and processing data from data base.

• Package Exceptions contains custom exception classes for project logic.

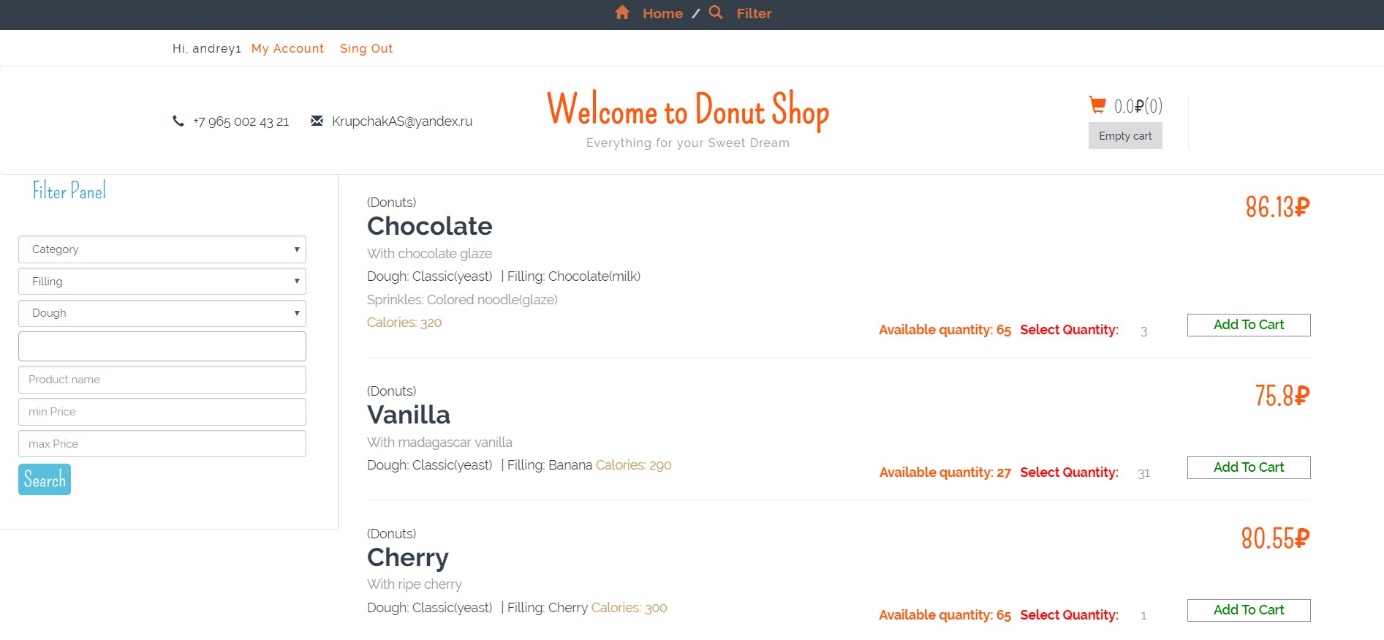
• Package Dao contains classes that is used for communication with the database.

• Package Entities contains classes that describe project entities.

• Package Config contains class of JMS configurations.

**9 User Interface**

**9.1 Filter**



There is one filter in the application. It allows to find goods using such options as Category, Filling, Dough, Sprinkle, Product Name, Min Price and Max Price. All fields are not obligatory.

**9.2 Templates**

Application doesn’t contain any templates.

**9.3 Used libraries**

• Jquery;

• Jquery-ui;

• SimpleCart;

• Sweetalert;

• Bootstrap;

• Bootstrap multiselect.

**10 Business Logic**

Service layer contains parametrized common methods for all classes. Creating, updating, removing operations has annotation @Transactional with Propagation.REQUIRED.

CRUD operations for DAO checks existing (using different parameters) and searching (using fields and “like” fields):

• AddressService;

• CategoryService;

• DoughService;

• FillingService;

• OrderService;

• ProductService;

• SprinkleService.

It is used for operations with DAO and for security in application:

• SecurityService.

It is used for operations with DAO and for users in application:

• UserService.

**11 Entities and DAO**

**11.1 Entities**

Entities are located in the entity package.

It is designed for user entity project:

• User;

• Role is designed to describe roles. The application has 2 roles: User, Admin. User can buy products, Admin can do CRUD operation on products, categories, fillings, doughs, sprinkle, and analyze statistic and edit order statuses.

• Address.

It is designed for Product entity project:

• Product;

• Category;

• Dough;

• Filling;

• Sprinkle.

It is designed for Order entity project:

• Order;

• OrderProduct.

Order statuses and options:

• DeliveryOption;

• PaymentOption;

• PaymentStatus;

• OrderStatus.

* 1. **DAO**

GenericDao contains parametrized common methods for all DAO classes, and contains Create, Update, Remove operations.

Other methods are:

• getById(Id key): get entity by id;

• getAll(): get all entity rows;

• selectForUpdate(Integer id).

All queries are constructed using JPA Criteria API.

The following classes have special methods and use methods from GenericDao:

UserDao:

• getByLogin(String login): get user by login;

• getByEmail(String email): get user by email.

AddressDao:

• getAddressByUserId(Integer id): query to get list addresses by User id.

CategoryDao, DoughDao, FilllingDao SprinkleDao:

• getByName(String name): query to get category by name.

OrderDao:

• getOrdersByUserId(Integer Id): query to get list orders by user;

• getOrdersForMonth(): query for list orders in current month;

• getOrdersForWeek(): query for list orders in current week.

OrderProduct:

• getOrderProductByOrderId(Integer orderId): query to get get all orderProduct by order.

ProductDao

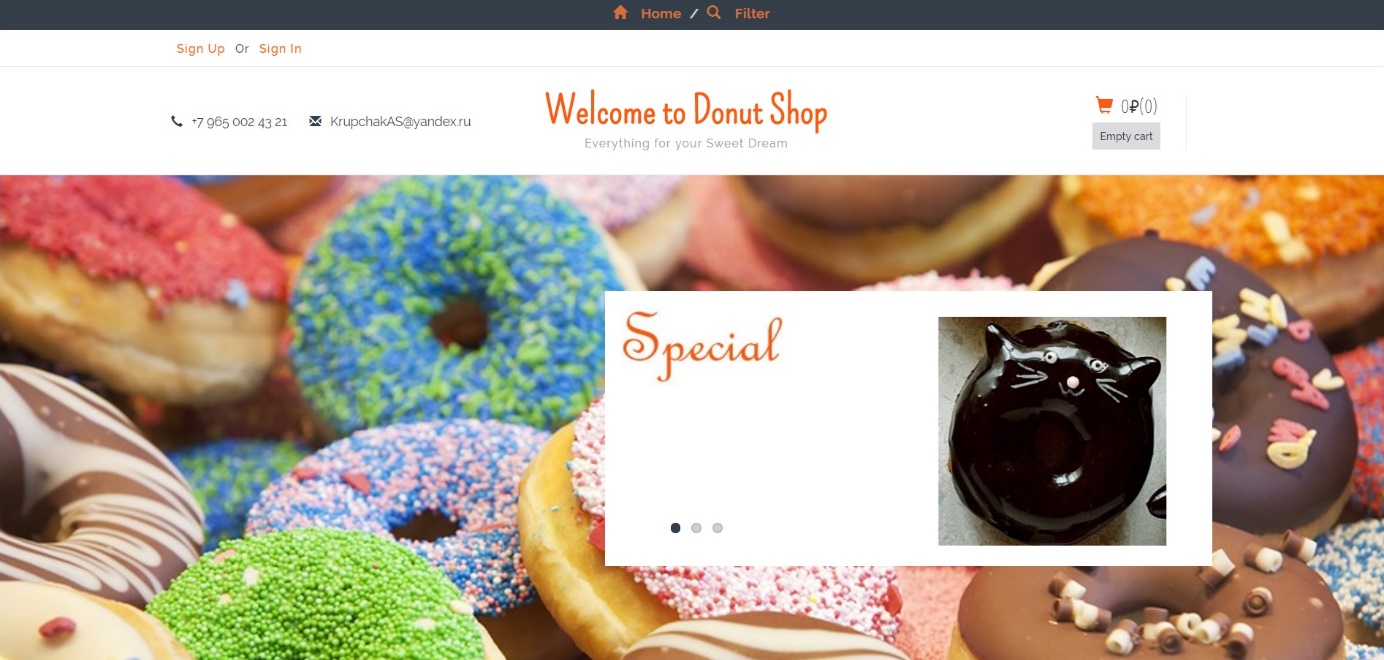
• getByName(String name): query to get category by name;

• getAllByCaterogy(Integer Id): query to get all product by category;

• getProductByParameters(FilterDTO filterDTO): query to get list product by fields object FilterDTO(categoryId, doughId, fillingId, sprinkleIdList, productName, minPrice, maxPrice).

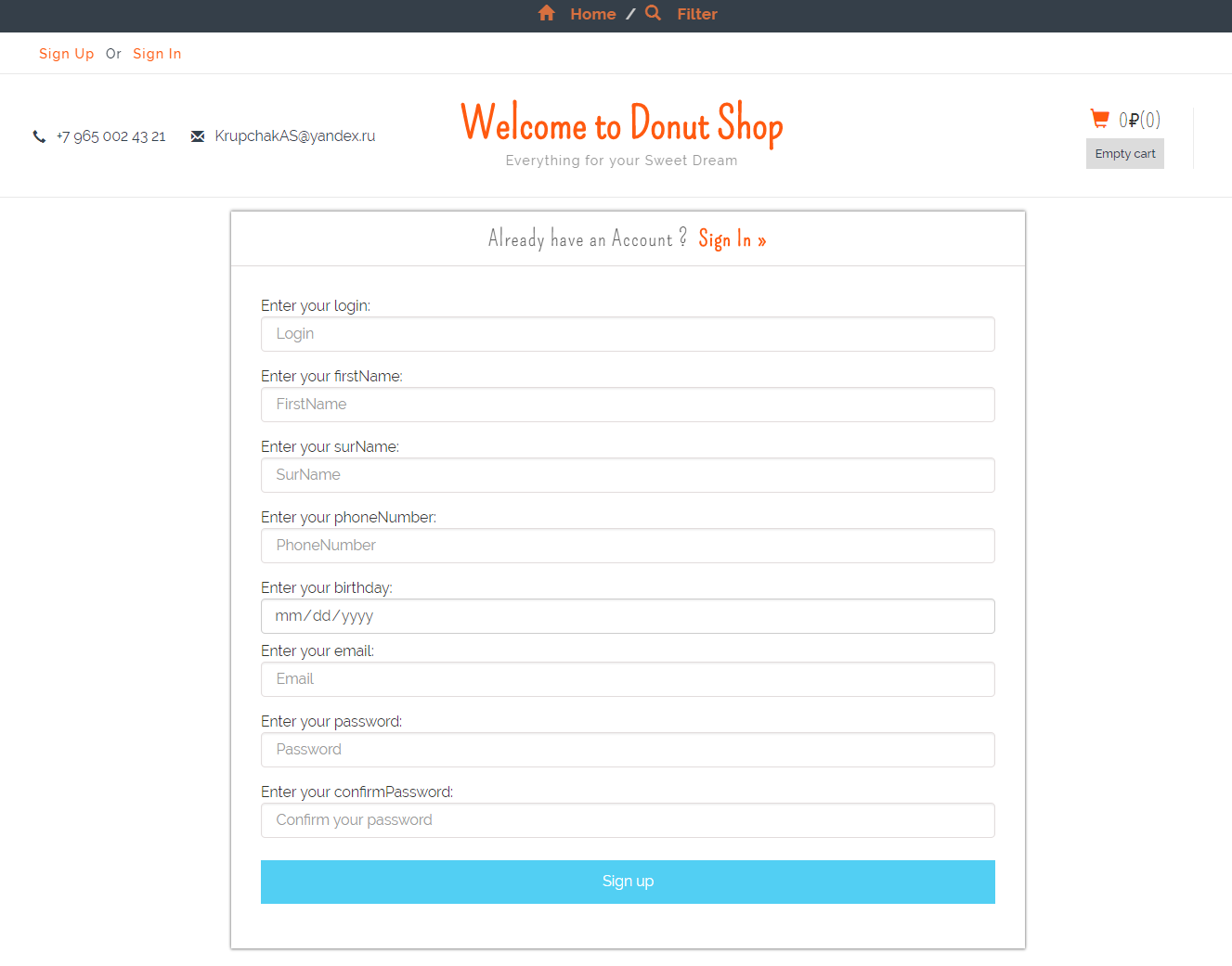
**12 Pages**

**12.1 Main Page**



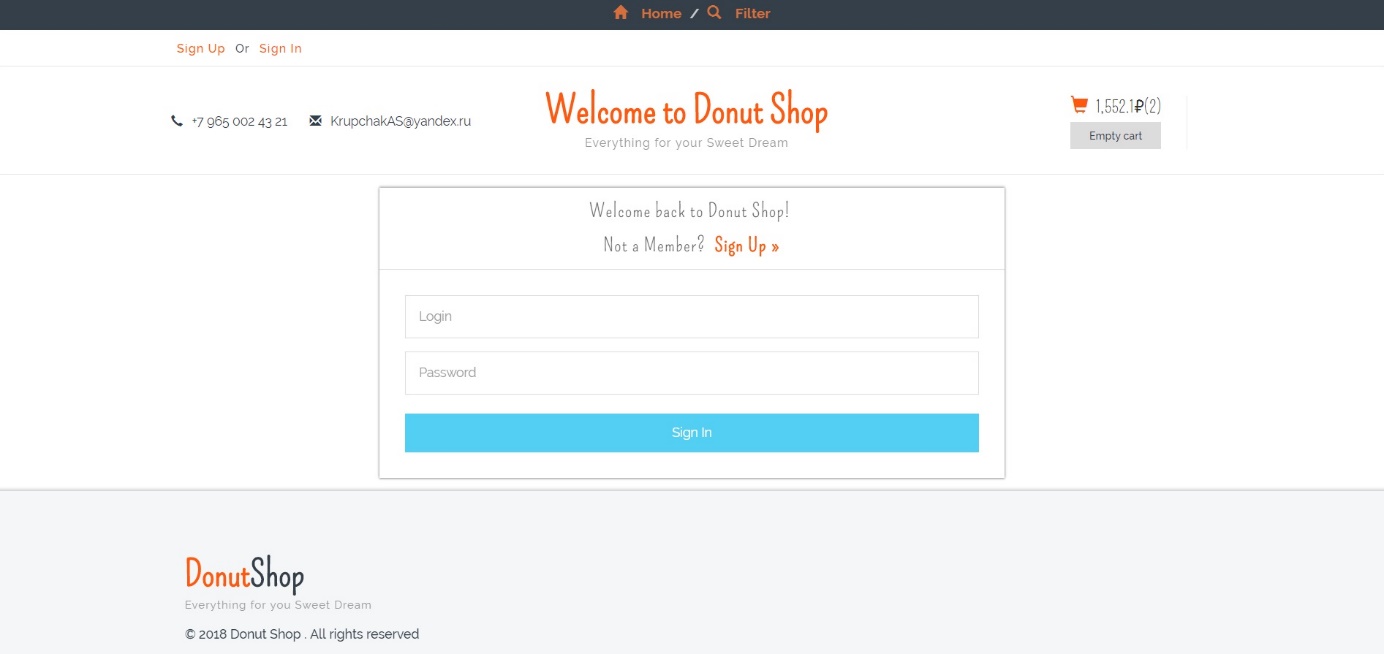
Main page contains contacts, links to account or registration (Sing Up or Sing in) and to filter. There is also an information about the cart and an opportunity to make cart empty.

* 1. **Registration**



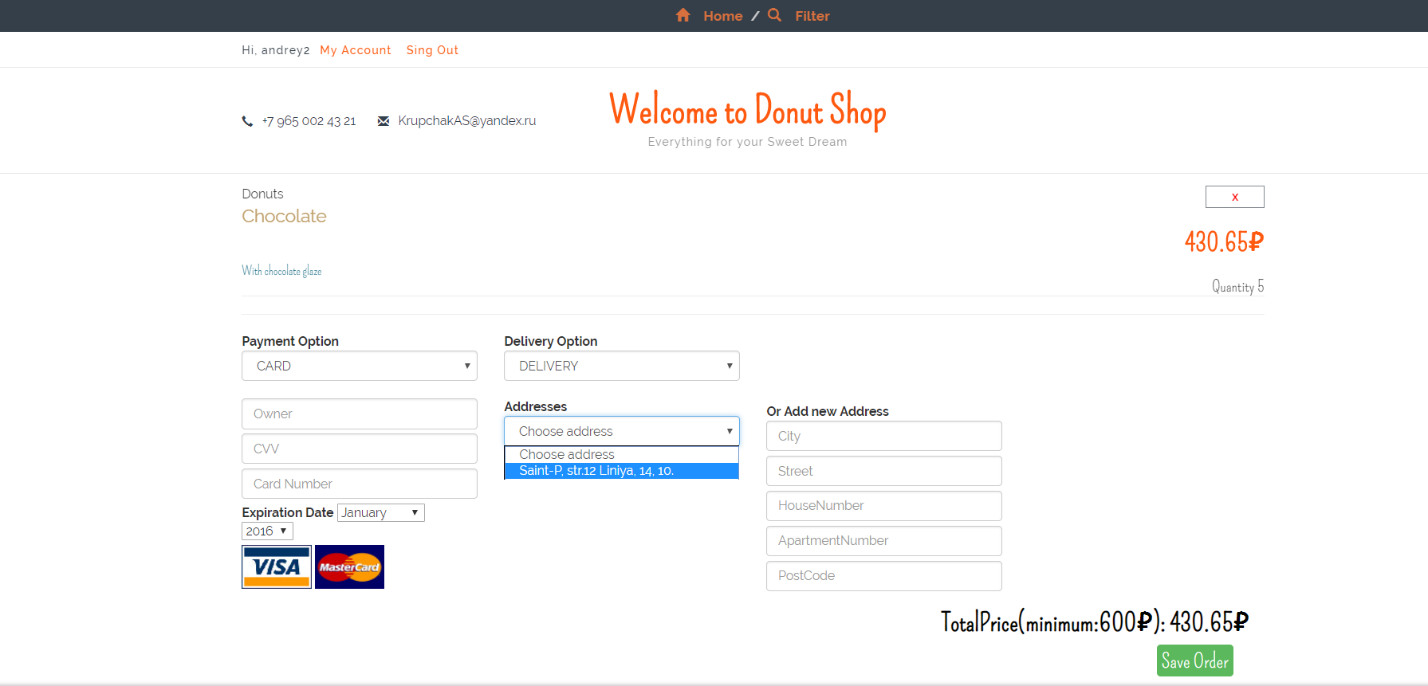
The page contains fields Login, First Name, Surname, Phone Number, Date of Birth, E-mail, Password and its confirming. If all fields are filled correctly, user is registered.

* 1. **Login Page**



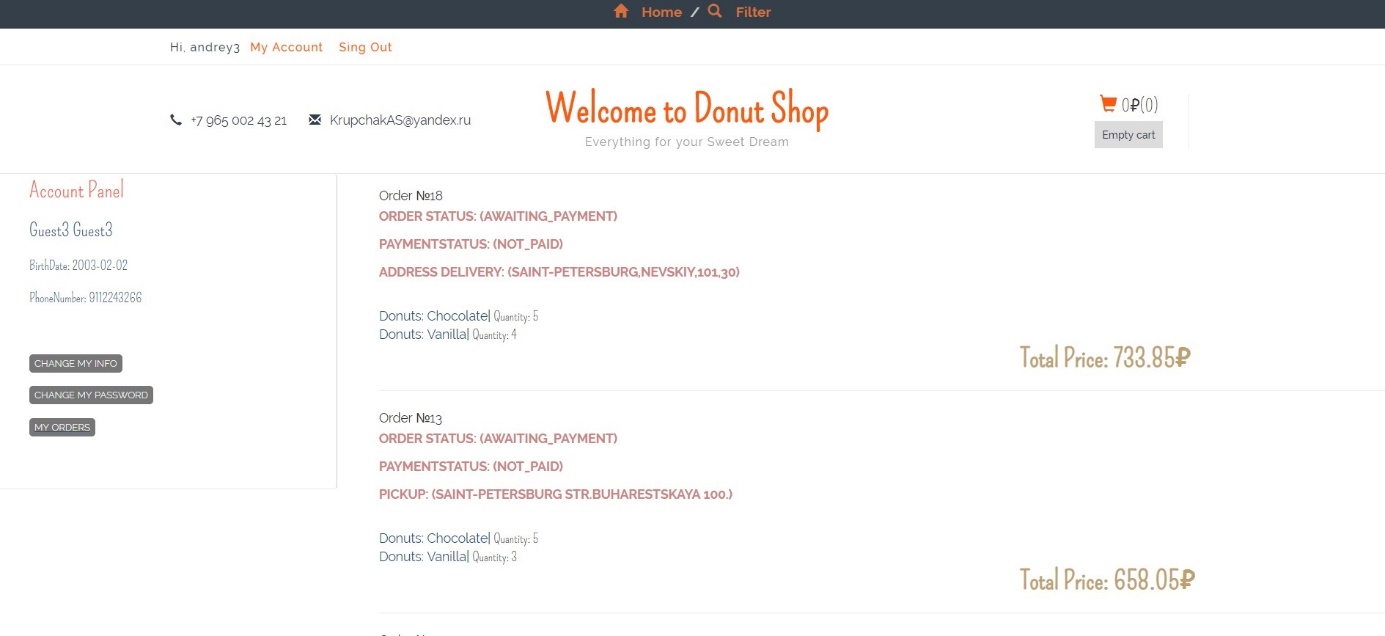
The page contains fields Login and Password. If all fields are filled correctly, user is singed in.

**12.4 Cart Page**



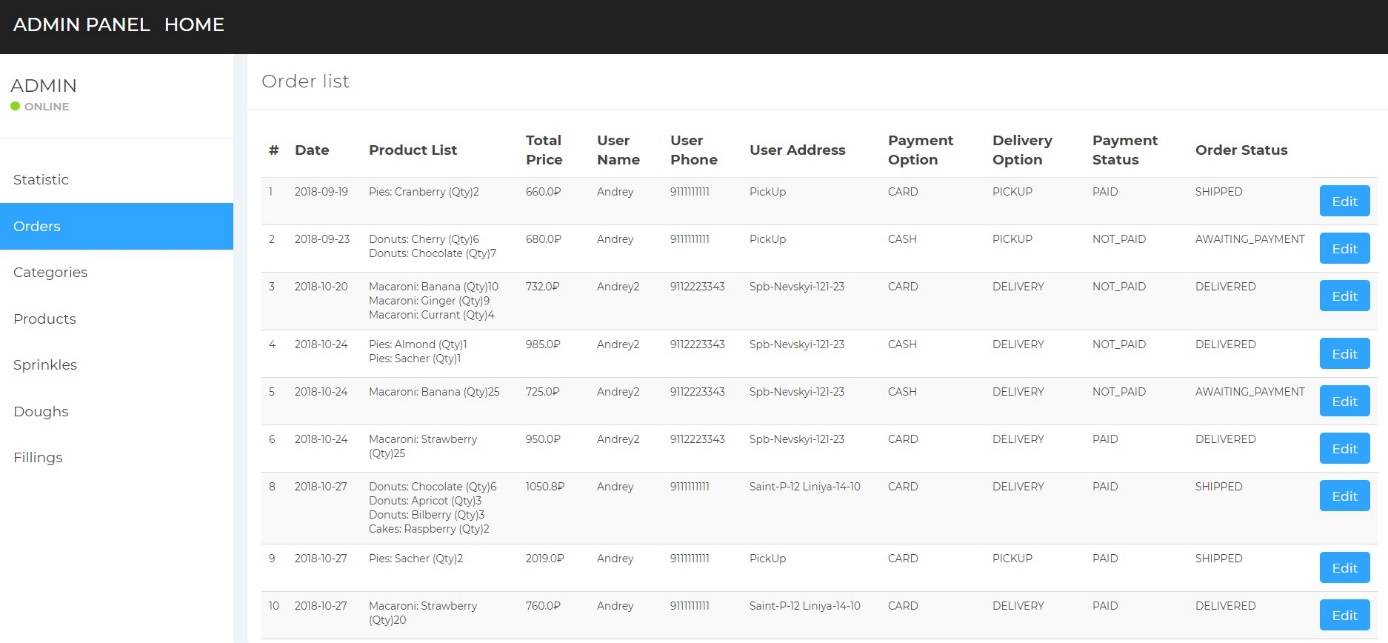
There is an information about the chosen goods with the quantity and prices and about the Total Price. There is also an opportunity to remove goods, to choose Payment Option (Card or Cash), to choose Delivery Option (Pick Up or Delivery), to choose address or to add a new one.

* 1. **My Account**



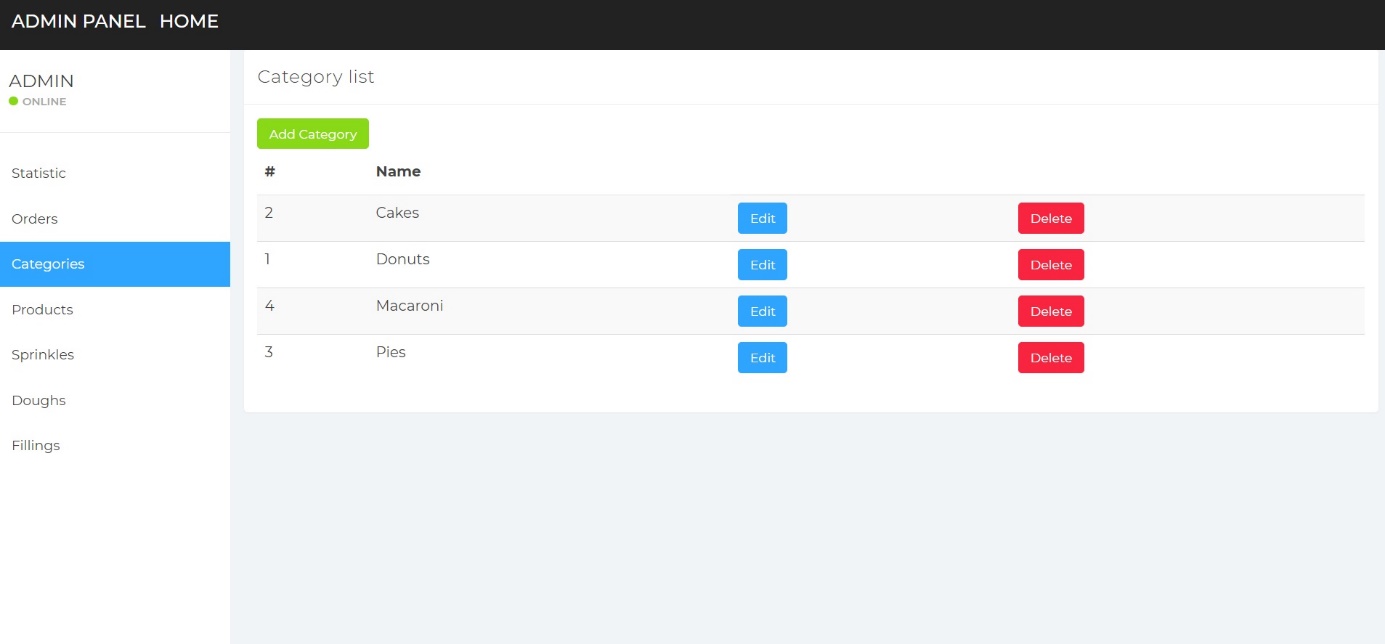
My Account contains user information and the information about all his/her orders. There is an opportunity to change user information (button “CHANGE MY INFO”), password (button “CHANGE MY PASSWORD”).

* 1. **Admin Page: Orders**



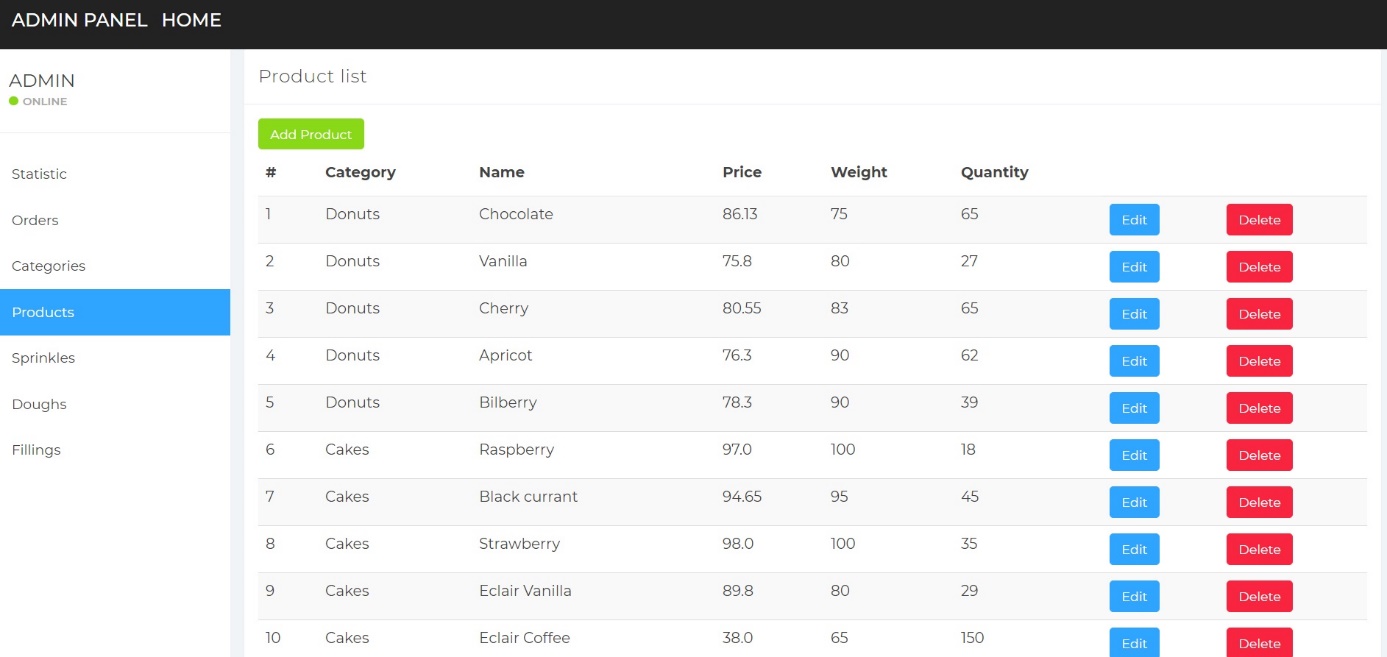
This page contains the list of all orders with such information as Date of Order, chosen products (Product List), Total Price, User Name who placed the order, User Address, Delivery Option, Payment Status and Order Status. There is also an opportunity to change the Order Status (button “Edit”).

**12.7 Admin Page: Categories**



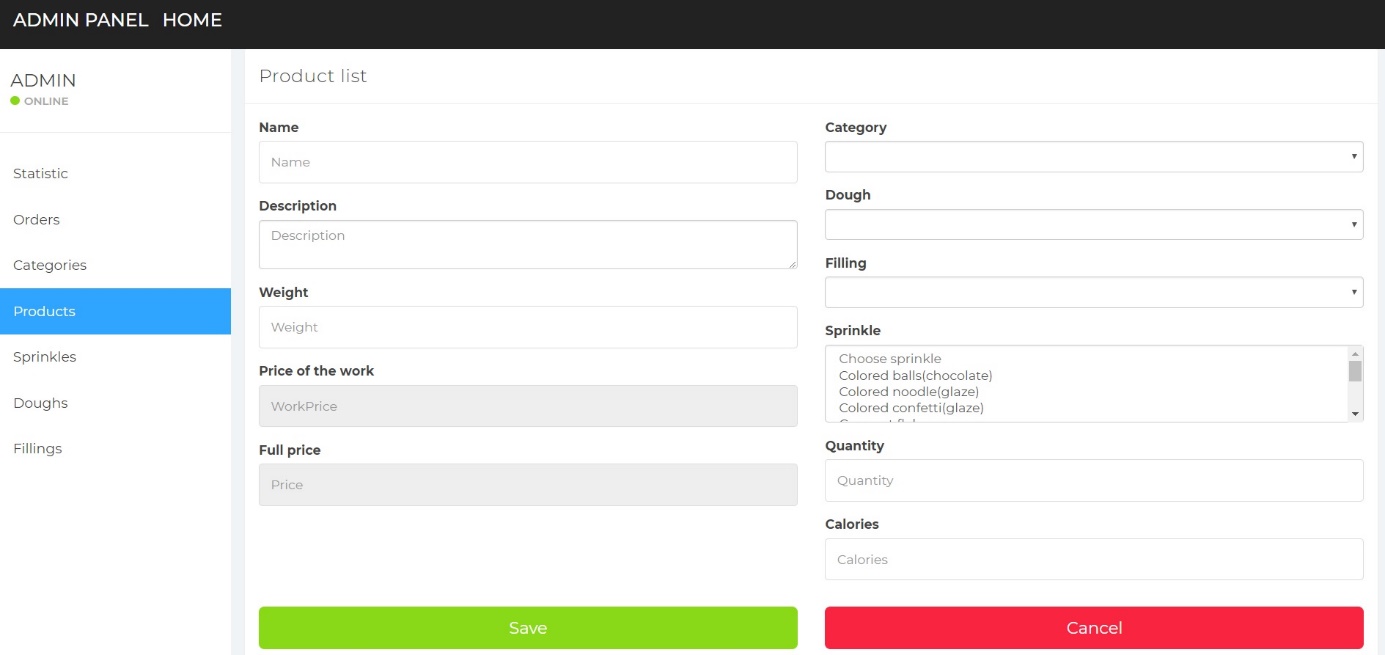
The page displays all created categories and allows to edit and to delete them. There is also an opportunity to add new categories.

* 1. **Admin Page: Products**



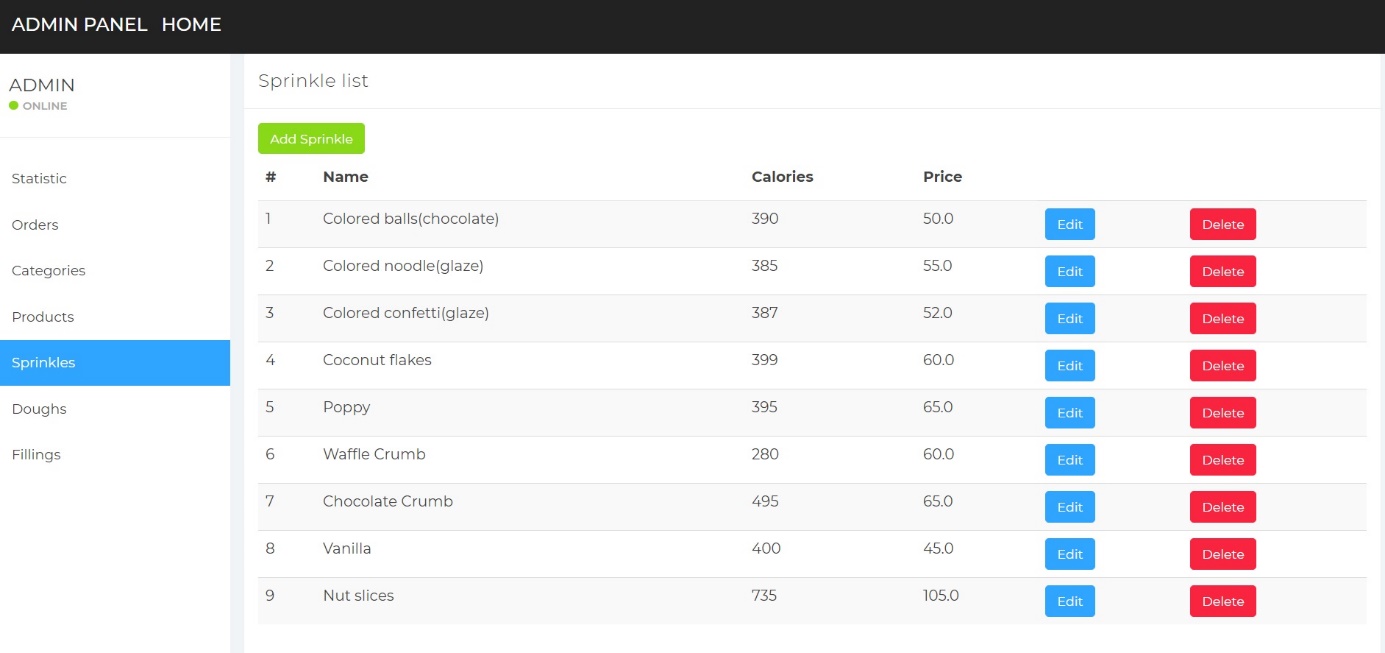
The page displays all created products with their category, price, weight, quantity and allows to edit and to delete them. There is also an opportunity to add new products.

* 1. **Admin Page: Products – Create or Edit**



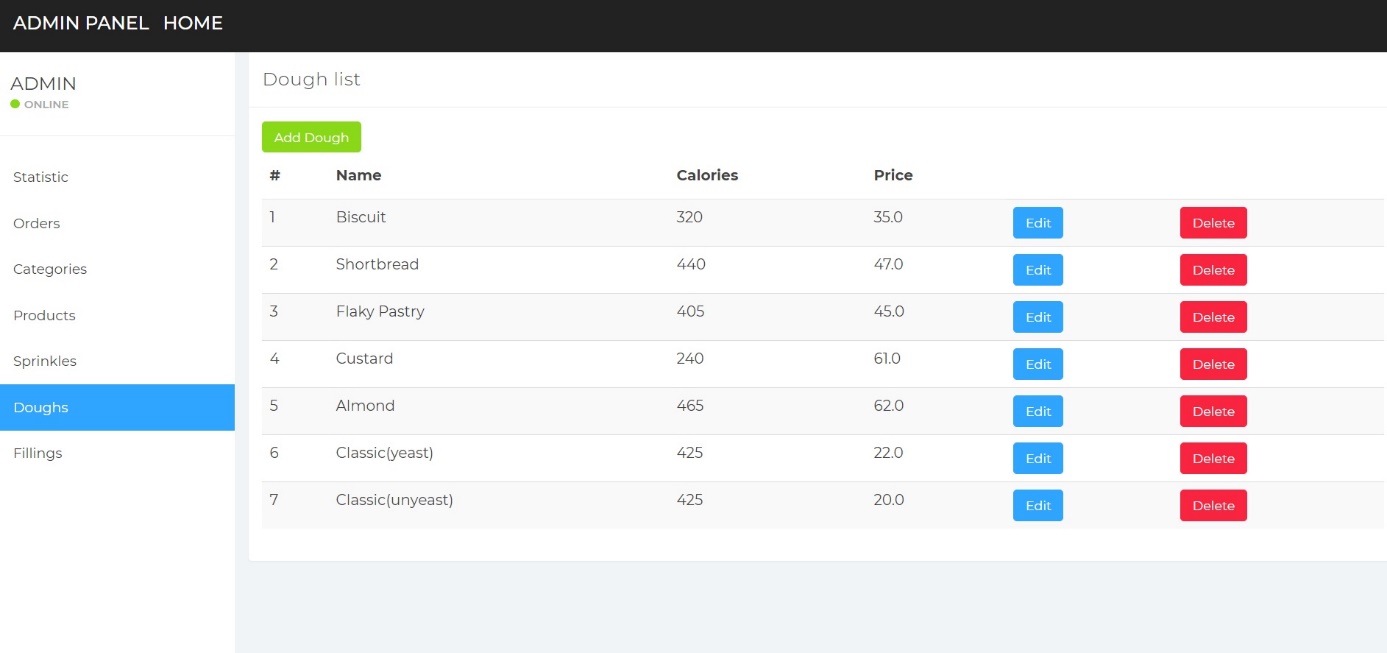
For creating or editing a product it is necessary to fill fields: Name, Description, Weight, Price of the work, Category, Dough, Filling (is not obligatory), Sprinkle (is not obligatory), Quantity and Calories. The field “Full price” is formed automatically according to chosen Price of the work, Dough, Filling and Sprinkle.

* 1. **Admin Page: Sprinkle**



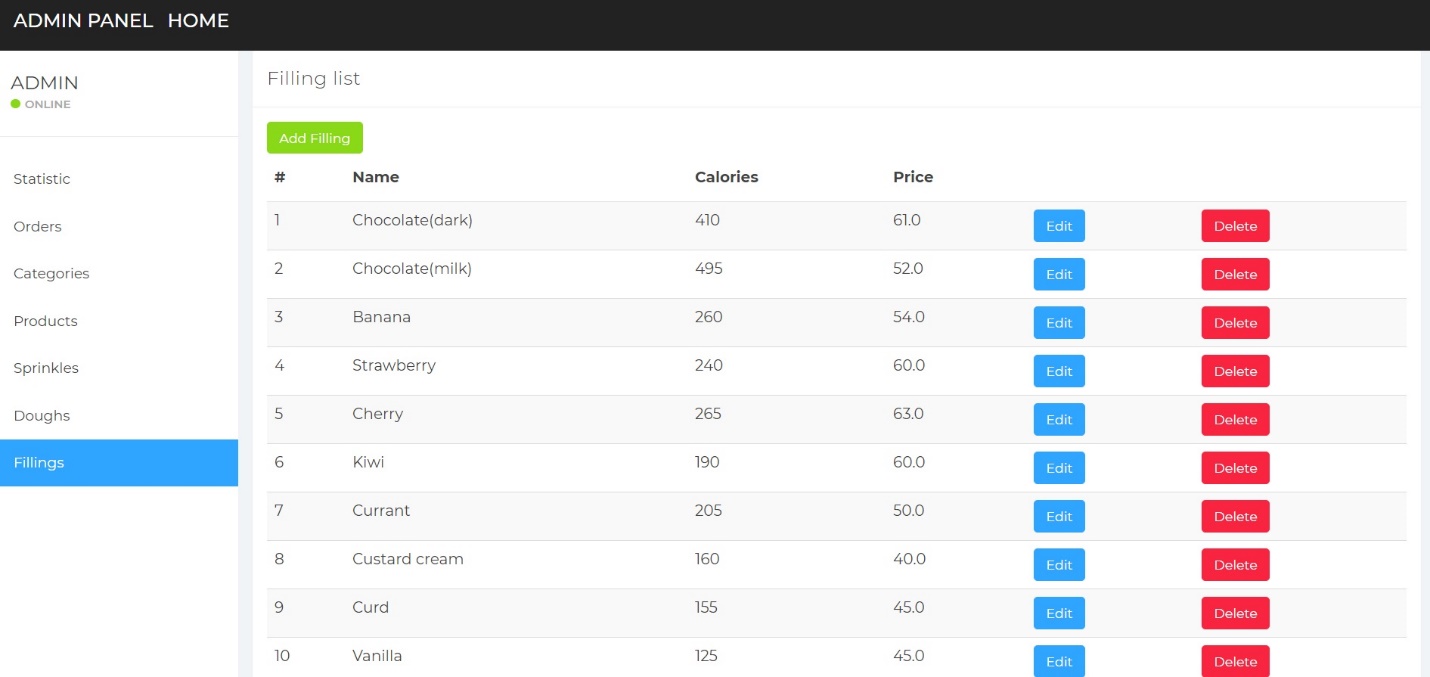
The page displays all created sprinkles and allows to edit and to delete them. There is also an opportunity to add new sprinkles.

* 1. **Admin Page: Doughs**



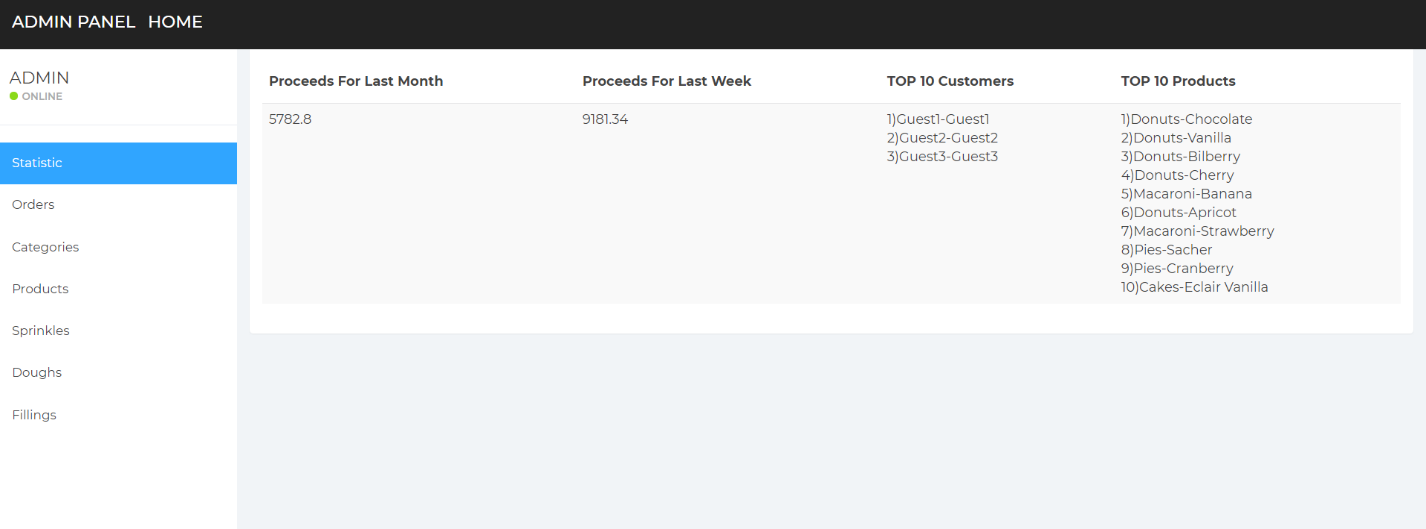
The page displays all created doughs and allows to edit and to delete them. There is also an opportunity to add new doughs.

* 1. **Admin Page: Fillings**



The page displays all created fillings and allows to edit and to delete them. There is also an opportunity to add new fillings.

* 1. **Admin Page: Statistics**



The page contains information about revenue for the month, revenue for the week, TOP 10 customers, TOP 10 products.

**13 Junit Tests**

• CategoryServiceMockTest test service CategoryService methods;

• DoughServiceMockTest test service DoughService methods;

• FillingServiceMockTest test service FillingService methods;

• SprinkleServiceMockTest test service SprinkleService methods;

• ProductServiceMockTest test service ProductService methods;

• OrderServiceMockTest test service OrderService methods.

**14 Ideas and Features**

• Downloading, keeping and using product images in remote Cloud.

Time for implementation and testing: 10 man-hour.

• Mailing with the information about the order.

Time for implementation and testing:8 man-hour.

• Forgot password?

After clicking “Forgot password?”, a modal window with three fields opens. These fields are e-mail, login and capture input field. After filling the fields, a user clicks the button. If input data matches data in the database and capture is correct, e-mail that is generated using SecureRandom class set of characters is sent to a user.

Time for implementation and testing:10 man-hour.

• Authorization via social networks.

Time for implementation and testing: 15 man-hour.