

# Uliège eShop

## e-commerce and e-business project

Beguin Mathias - Lejoly Loic

Github access: <https://github.com/MTHSBGN/e-commerce>

November 2018

# 1 Introduction

For this project, we decided to work on the proposition given in the statement (Uliège eShop). As asked in the statement the goal is to provide a website where objects related to the University can be sold. To achieve that, we decided to use a relational database to store products and data related to customers. Concerning the back-end's website *Node.js* and SQL are used. *Node.js* is a cross-platform JavaScript run-time environment. That means JavaScript is used outside client browser. This cross-platform is young compare to PHP but more and more companies tend to use it due to its design and its scalability. Concerning the front-end part as usual HTML, CSS and JavaScript are used.

## 2 Database architecture

To model the database of the Uliège eShop website a relational database architecture was chosen. The reason of this choice is justified by the product catalogue is not large and because it can be hard to have a good NO-SQL design. On the web you can find some debates that discuss the fact to use relational DBs or no-SQL DBs. The advantage of the no-SQL DB is that it is easier to manage your product catalogue (product variations) but the drawback of no-SQL DBs is the reliability (can be solved with DB clustering e.g: foundationdb, couchDB,...). Concerning the reliability relational DB architectures are, for the majority, ACID compliant. These two types of architectures are not mutual exclusive. Most of the time, e-commerce websites use of combination of SQL and No-SQL DBs (a paper discussing that topic <https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1026&context=sais2013>).

### 2.1 entity-relationship model

Figure 1 depicts the entity-relationship model based on the minimal requirements given in the statements. We decided to split the product table proposed in the statement into multiple tables to avoid data replications and to provide a scalable approach. Indeed, in the product table proposed, all data linked to a product is packaged into this table. The result of that is it is difficult to manage products with different properties (e.g colors, size,...) without replications. That is why we decided to split this table into several tables that are *Sku* (stock keeping units) *Variant* (gives the properties of a product), *Image* (image paths related to a product), *Description* (Give the product description), and *Category* (specify the product category).

Concerning the transaction part we decided to split that part into two related steps. The first step establishes a relation between the sku ID, the order ID (defined in the second step) the current price of the product during the order and the quantity needed. This step is achieved with the table *Order\_details*. The second step consists in summarizes the the order of the client by referencing the ID of the order, the delivery address, the total price of the order, and the user ID.

### 2.2 Relational model

1. Customer(client\_id, mail, username, password, delivery\_address, firstname, lastname, cookie\_id)
2. Client\_order(order\_id, #client\_id, date, total\_price, ship\_address)
3. Order\_details(#order\_id, #sku\_id, quantity, price)
4. Product(product\_id, #category\_id, #image\_id, #description\_id, name)
5. Image(image\_id, #sku\_id, filename)
6. Sku(sku\_id, #produt\_id, price, available, sold)
7. Variants(variant\_id, attribute, value)
8. Description(description\_id, french, english)
9. Category(category\_id, #description\_id, name)

### 2.3 model to SQL tables

The entity-relational model depicted on Figure 1 can be translated into SQL tables as Figure 2 shows. If you want a more interactive view of these tables, you can load the SQL file in *phpmyadmin* for instance.

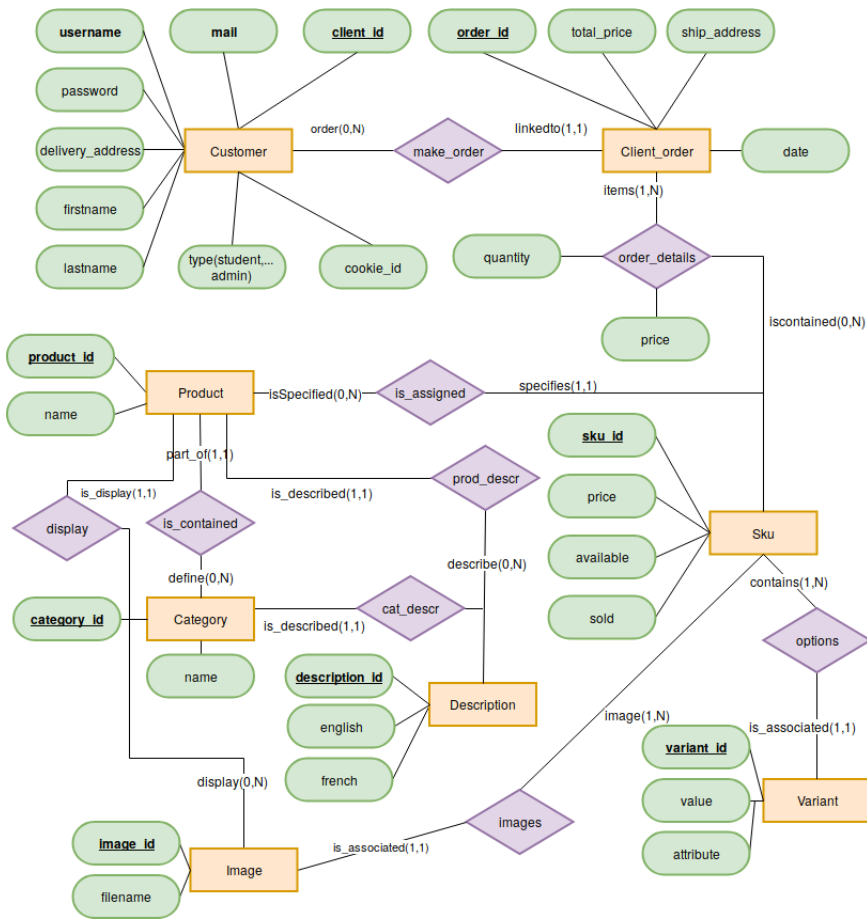


Figure 1: eShop entity-relationship model

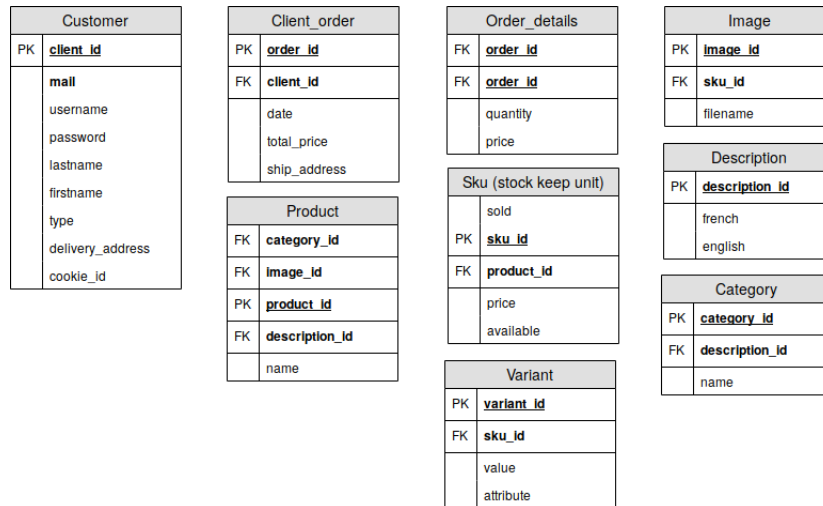


Figure 2: eShop entity-relationship model

### 3 Website

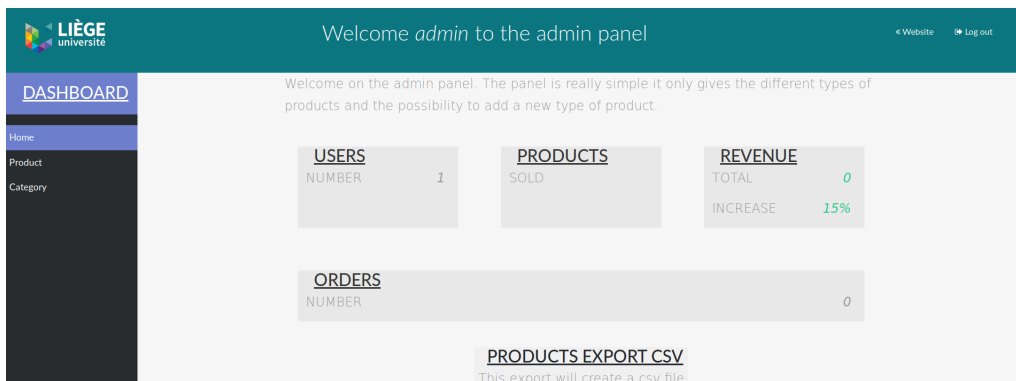
The website is divided into section. The first section is dedicated to the admin of the website itself. This panel can only be accessed by users with the right privileges. The second second is dedicated to customer where they can buy products. For the development of the website we applied minimal protections like SQL injection managed by the *Node.js* server.

### 3.1 Admin panel

The admin panel is a dashboard provided to the administrator of the website. The users that can use this panel have special privileges. To detect this kind of users the field *type* of customer table is set to zero. As results, a admin user needs to follow the regular registration phase to create an account and then the super user that has a direct access to the database can change the *type* field to promote a user as admin user. Admin users have an extra button to allow them to access to the admin panel.

The dashboard is simple, at left a menu can help the user to navigate through the different webpages. As you can see on the Figure 3 the panel only contains 3 pages (*home, product, category*). The homepage summarizes different parameters, you can also export some data. The product and category pages give the possibility to add new elements to the database easily.

Even if you are on the dashboard you can go back to the e-commerce pages just by clicking on the website button.



(a) admin homepage

The screenshot shows the product manager form. It has a teal header with the text "Welcome admin to the admin panel". Below the header, there is a section titled "ADD A NEW PRODUCT DESCRIPTION" with a button "Add a new product description". This section contains input fields for "name", "-- select a product --", "french description", and "english description", followed by an "add" button. Below this is another section titled "ADD A NEW SKU FOR EXISTING PRODUCT" with a button "Add a new sku". This section contains a dropdown menu for "-- select a product --", a text input field for "Not already used", and a text input field for "sku id".

(b) product manager

Figure 3: admin pages

### 3.2 Customer View

If you are a regular customer, you can just have access to the e-commerce webpages. On the homepage you can see the different products that are sold by the University. On top of the website at right you have several buttons. One to create an account, one to have access to your basket and one which is the about page. To buy

products a user needs to create an account and this account must be active.

A user can scroll the homepage that contains all products and when he is interested by a product, he can click on that product and he will be redirected to the detailed product page.

Like common e-commerce websites, users can search for products by using search-bar or simply by walking through the website. Concerning this website, we do not receive enough information concerning products (i.e: categories, etc.) so we decided to display categories since the number of products to display is rather small. Nonetheless, products have a category as you can see on the database schema. Administrators have thus the choice to use it or not.

### 3.2.1 Registration

To purchase products, a customer needs to create an account on the website to do so, he needs to click on the button dedicated to that purpose or simply follow the appropriate url. On that page the user must fill fields with correct informations and then validate his account. If everything is correct, he is redirected to the login page.

### 3.2.2 Log in

After the creation of an account, a user can connect to the website by using his account information. To achieve that he simply clicks on the button made for that purpose or is going to the appropriate url. If information given is correct he will be redirected on the homepage.

The figure displays two web forms side-by-side. Form (a) is titled 'Inscrivez-vous' and contains input fields for 'Prénom', 'Nom de famille', 'Adresse email', 'Nom d'utilisateur', 'Mot de passe' (twice), and 'Adresse de livraison'. It also features a dropdown menu labeled 'Choisissez une option' and a teal 'S'inscrire' button. At the bottom, it says 'Déjà inscrit ? [Connectez-vous](#)'. Form (b) is titled 'Connectez-vous' and has input fields for 'Nom d'utilisateur ou email' and 'Mot de passe', followed by a teal 'Se connecter' button. At the bottom, it says 'Pas encore inscrit ? [Créer un compte](#)'.

(a) register page

(b) login page

Figure 4: connection pages

The login page is the same for the admin, since the admin has more privileges when he is logged, he can access more pages such as the admin panel.

## 3.3 Basket

To store the basket of a user, HTTP cookies are used. When a user clicks on its basket, he obtains a summary of his products. This summary gives the possibility to modify product quantities or to delete some of them. When the user is in accordance with his order summary, he can validate it only if he has an account and if he is connected. The validation process sends information to the payment procedure where all elements are stored and where the order becomes effective.

### **3.3.1 Payment**

Since none payment procedure like PayPal etc. is set for this project, a simple payment page is used to update the database. Even if it is a simple page, a transactional commit phase is established to be sure that all tables are updated correctly.

## **3.4 Security aspects**

Concerning the security of the website, some common guidelines are applied such as SQL injections and transactional commits. But to provide a robust website some other security rules must be applied such as directory permissions, redirection rules by catching different errors,...

## **4 Market model**

Concerning the target of the website it is obvious. It concerns essentially Alumni, employees and students of ULiège. The eShop provides a bunch of goodies related to the university and the city of Liège. Prices given are affordable for student and the aim of the eShop is not to be competitive with others websites that sell similar products. The aim is to provide a website where university community can buy university products easily. The website can also be used by student associations to sell their pulls for instance and provides a legal framework to sell them.

## **5 Improvements**

Concerning the website some improvements can be done. First of all concerning the design, we tried to have a flat design user interface but we are not web designer nor user experience designer (UX). After that, security aspects must be improved and not neglected (DDoS attacks,XSS,etc.).