



# Web Project

---

*Architecture web/ Traitement Données relationnelles*

22/01/2014

# Sommaire

<b>A</b>	<b>Présentation .....</b>	<b>Erreur ! Signet non défini.</b>
A.1	Sujet.....	<b>Erreur ! Signet non défini.</b>
A.2	Organisation .....	<b>Erreur ! Signet non défini.</b>
<b>B</b>	<b>Choix techniques.....</b>	<b>Erreur ! Signet non défini.</b>
B.1	Architecture technique .....	<b>Erreur ! Signet non défini.</b>
B.2	Architecture de déploiement .....	<b>Erreur ! Signet non défini.</b>
<b>C</b>	<b>Ressenti personnel.....</b>	<b>Erreur ! Signet non défini.</b>
C.1	Difficultés rencontrées .....	<b>Erreur ! Signet non défini.</b>
C.2	Points positifs .....	<b>Erreur ! Signet non défini.</b>

# A Présentation

## A.1 Sujet

Le sujet de ce projet Web a été de créer une application web de A à Z, frontend et backend.

Le choix de l'application fût libre et j'avais donc mon idée en tête quant à l'application que j'allais réaliser. En effet, l'application créée nait d'un besoin réel : pouvoir consulter le menu du restaurant universitaire sur internet est quelque chose qui me semblait évident et pourtant aucune plateforme ne délivrait ce service.

C'est pourquoi j'ai décidé de créer une plateforme web, permettant de consulter les menus des différentes salles du restaurant universitaire Triolet. De plus, cette application est faite pour être utilisée par les administrateurs pour ajouter les menus au jour le jour.

Une fonctionnalité intéressante que j'ai voulu implémenter est le fait que les utilisateurs peuvent donner leurs avis sur la qualité des repas, notamment à travers de commentaires et de « like ».

A la découverte de ce projet, j'ai ressenti beaucoup d'appréhension étant donné que je ne suis pas expérimenté dans le développement d'application web.

URL du site web : [checkmymenu-restou.rhcloud.com](http://checkmymenu-restou.rhcloud.com)

## A.2 Organisation

Etant donné mes faibles connaissances dans la matière pour ce projet, je me suis laissé conseiller par des camarades plus expérimentés. En effet, ils m'ont conseillé d'utiliser Wamp Server pour démarrer mon projet en local. Ce logiciel est vraiment intéressant car il permet d'utiliser une base de donnée phpMyAdmin gérée par MySQL.

Ne connaissant pas les autres langages de traitements de données comme PostgreSQL, je me suis lancé avec MySQL, qui ressemble à Oracle (vu en début de semestre). Pour accéder à cette base de donnée à travers mes pages web, le langage PHP m'est apparu comme évidence pour plusieurs raisons : c'est un langage qui peut s'assimiler assez rapidement et en outre, de nombreux cours sur internet sont disponibles (OpenClassroom, W3Schools).

Pour coder, j'ai principalement utilisé l'éditeur SublimText qui se révèle très fonctionnel, et est très agréable à utiliser.

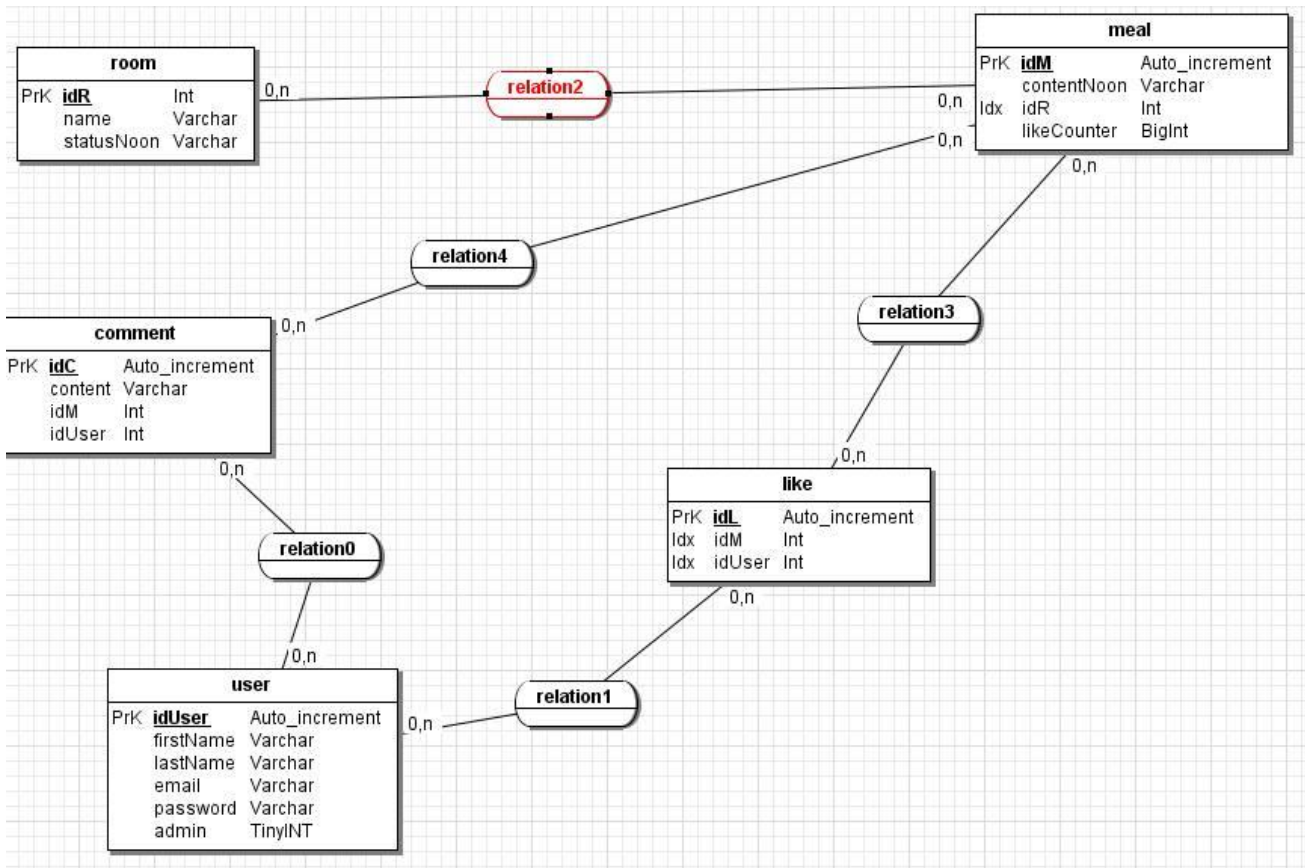
Pour finir, dans le cadre du certificat international et de notre formation fortement tournée vers l'anglais, j'ai essayé d'utiliser l'anglais le plus possible.

## B Choix techniques

### B.1 Architecture technique

**Gestion des données :** ma base de donnée est une base hébergé sur phpMyAdmin et traité avec le langage MySQL. Cette base comporte 5 tables : user, room, meal, comment et like.

Voici ci-dessous la struture de ma base de donnée :



J'ai utilisé un seul trigger (contrairement aux consignes en demandant deux) dans cette base de donnée, concernant la table like, qui consiste un incrémenter le compteur de like des menus à chaque fois.

**Architecture MVC :** j'ai essayé d'organiser mon projet sous l'architecture MVC (Model-View-Controller) qui permet de structurer son code en 3 catégories. Cette architecture nous a été présenté en cours et après recherche, m'a semblé une bonne option.

Cependant, cette architecture fût difficile à respecter étant donné le peu de formation à notre disposition.

En résumé, pour afficher une page, nous faisons appel au contrôleur, qui lui, récupère les données de la base à travers le modèle (en utilisant des requêtes SQL). Une fois les données récupérées, le contrôleur effectue des opérations (calculs), et renvoie la vue, qui affiche la page.

**Sécurité :** Concernant la sécurité de mon site, celle-ci a été la plus difficile à gérer car très peu d'information sur quelles options choisir pour protéger mon site.

J'ai choisi d'utiliser des cookies pour l'authentification d'utilisateur, en opposition aux sessions, car c'est une solution simple et sûre si on l'utilise correctement.

## **B.2 Architecture de déploiement**

Pour effectuer le déploiement de mon site sur le web, mon choix s'est tourné vers OpenShift. Lors de nos cours en architecture web, il nous a été présenté deux environnements : Heroku et OpenShift. J'ai choisi ce dernier car la plupart de mes camarades l'avaient déjà utilisé et n'étaient tombés sur aucun problème. De plus, je savais qu'ils pourraient m'aider en cas de problème.

Pour mettre le site en ligne, il m'a fallu installer Ruby afin d'installer RHC pour déployer le site sur les serveurs d'OpenShift. Une fois mon répertoire GitHub lié, je n'avais plus qu'à « add », « commit » et « push » pour continuer le développement du site.

## **C Ressenti personnel**

### **C.1 Difficultés rencontrées**

Lors de ce projet, les principales difficultés rencontrées ont été celles liées au temps.

En effet, deux semaines pour mener à bien un projet tel que celui-là, étant débutant comme moi, fût un délai bien trop court.

J'ai mis beaucoup de temps à assimiler les bases des langages utilisées ce qui m'a beaucoup pénalisé pour rendre un projet complet.

En effet, mon projet est incomplet avec plusieurs fonctions importantes manquantes (ajout de repas, modification de statut d'une salle). En plus de cela, un trigger me manque pour respecter les consignes. Tout cela me pénalise, mais je me rend compte des lacunes accumulées pour apprendre de nouveaux langages.

### **C.2 Points positifs**

Cependant, ce projet a été très enrichissant et m'a permis de m'apprendre à travailler sous pression.