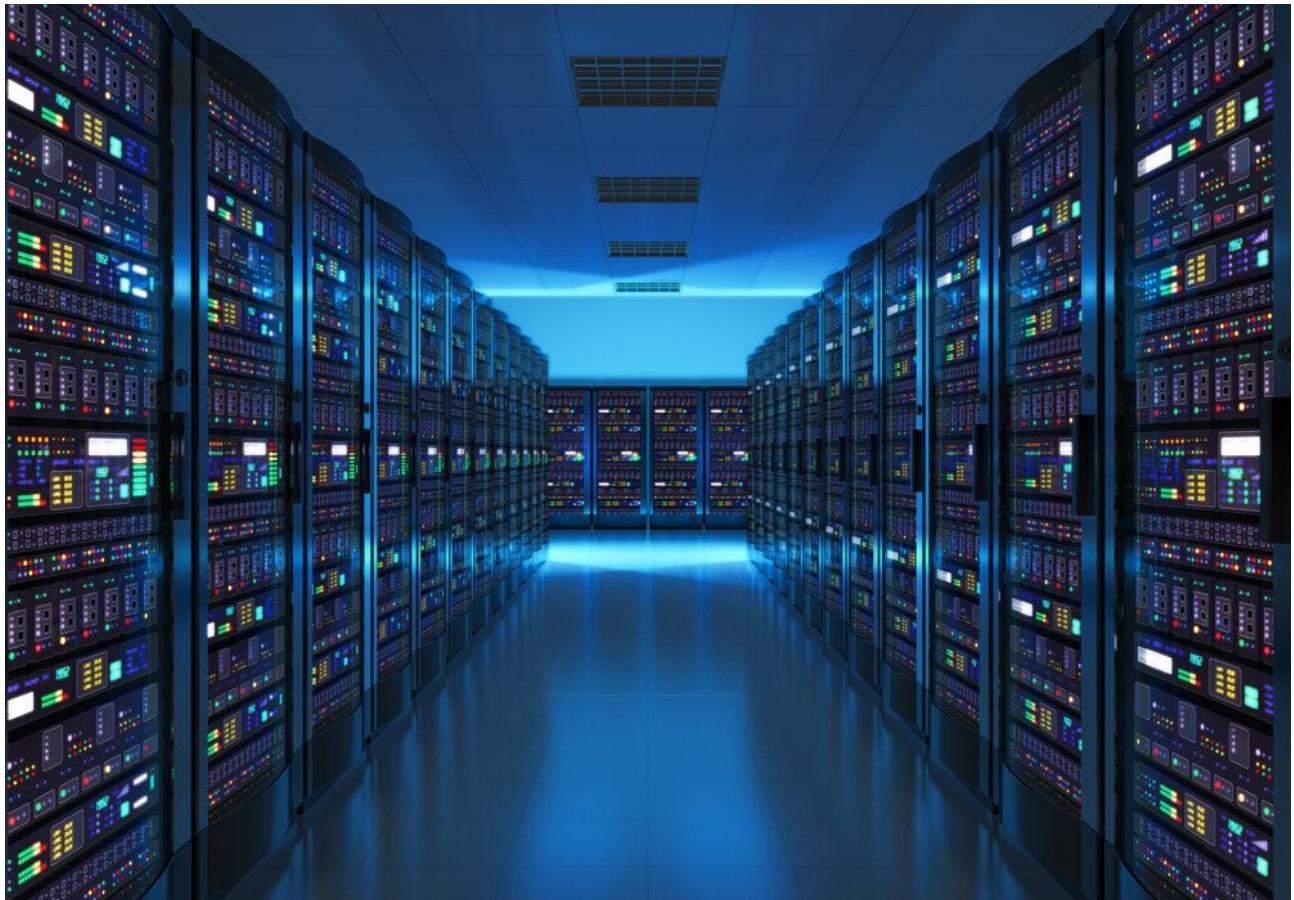


Nelson Camara
Pierre Le Veillo
Mezouar Mikael

Projet Base de Données :



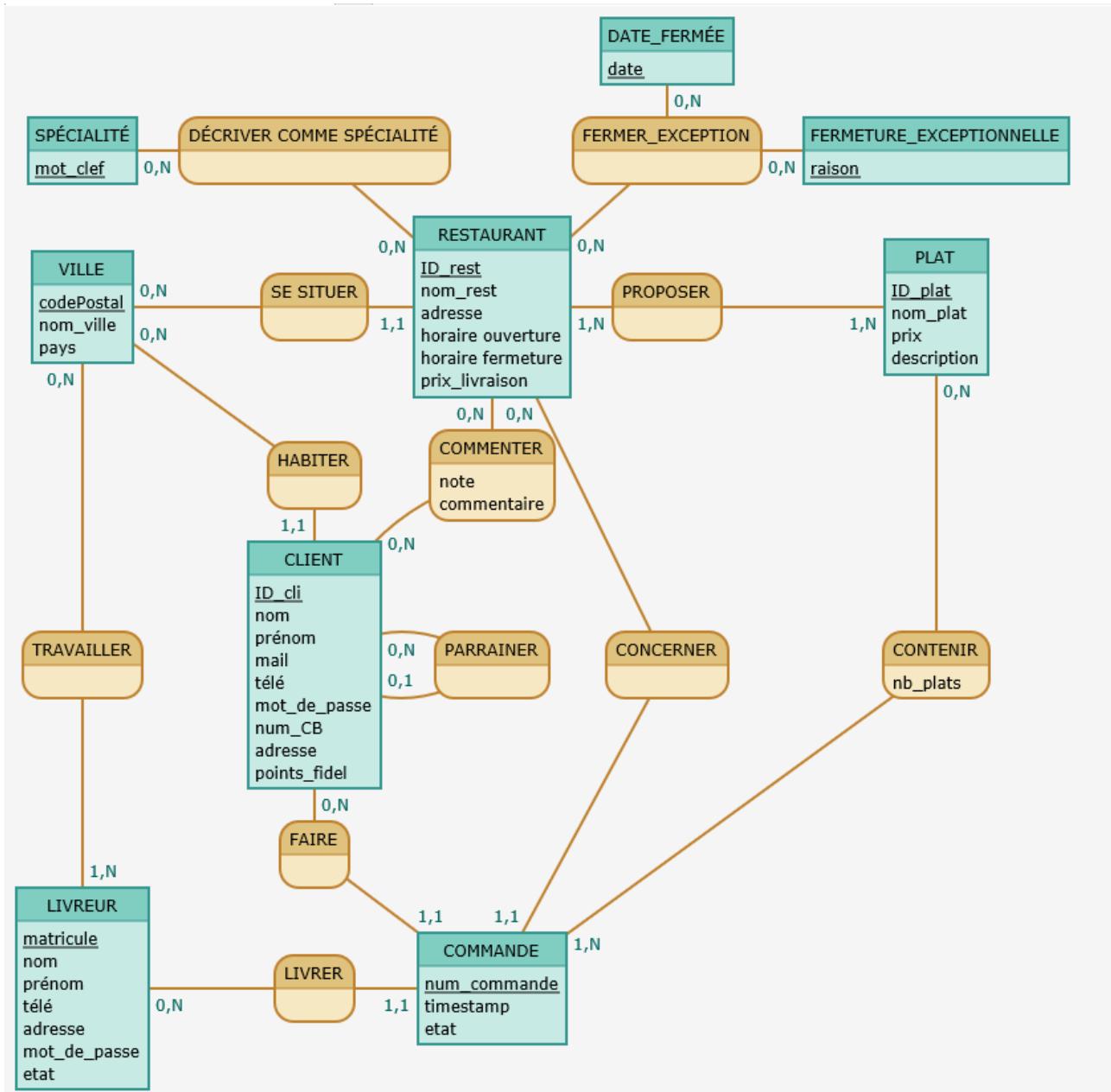
1)Présentation

Présentation du projet :

Enjoy est une plateforme de livraison de plats à domicile , dont le but est d'offrir à ses utilisateurs un vaste choix de restaurants et aux restaurants partenaires de ne pas avoir à gérer leur propre service de livraison . Ainsi , Enjoy ne vend pas directement de nourriture, mais se contente de transmettre aux restaurants les commandes passées sur la plateforme , et met ensuite à disposition des restaurants sa flotte de livreurs afin d'acheminer les plats préparés jusqu'au domicile des clients .

2) Section technique :

Schéma entité association :



Voici notre schéma relationnel construit à partir du schéma E/A du rendu 1 :

Restaurant (ID_rest, nom_rest, adresse, horaire_ouverture, horaire_fermeture, prix_livraison)

Client (ID_cli, nom, prénom, mail, téléphone, mot_de_passe, num_cb, adresse, point_fidel)

Livreur (matricule, nom, prénom, téléphone, adresse, mot_de_passe, état)
 Ville (code_postale, nom_ville, pays)

Plat (ID_plat, nom_plat, prix, description, ID_rest)

Commande (num_commande, timestamp, état, matricule, ID_cli, ID_rest)

Spécialité (mot_def, ID_rest)

Fermeture_exceptionnelle (date , ID_rest, raison)

Commenter (ID_cli, ID_rest, note, commentaire)

Contenir (num_commande, ID_plat, nb_plat)

ID_cli dans Commande fais référence à ID_cli dans Client

ID_rest dans Commande fais référence à ID_rest dans Restaurant

matricule dans Commande fais référence à matricule dans Livreur

ID_cli dans Commenter fais référence à ID_cli dans Client

ID_rest dans Commenter fais référence à ID_rest dans Restaurant

ID_plat dans Contenir fais référence à ID_plat dans Plat

num_commande dans Contenir fais référence a num_commande dans Commande

ID_rest dans Plat fais référence à ID_rest dans Restaurant

Voici à présent les contraintes qu'imposent nos schémas :

- un livreur peut livrer 1 et 1 seule commande
- 0 ou plusieurs clients peuvent faire 1 et 1 seule commande
- un restaurant ou plusieurs resto propose 1 ou plusieurs plat
- un livreur ou plusieurs livreurs travaille dans une ou plusieurs ville
- un client habite dans 0 ou plusieurs villes
- 0 ou un clients parrainent 0 ou n clients
- 0 ou plusieurs clients commentent 0 ou plusieurs restos
- 0 ou plusieurs restos décrivent 0 ou plusieurs spécialités
- 0 ou plusieurs restos ferment à 0 ou plusieurs dates et pour 0 ou plusieurs raisons

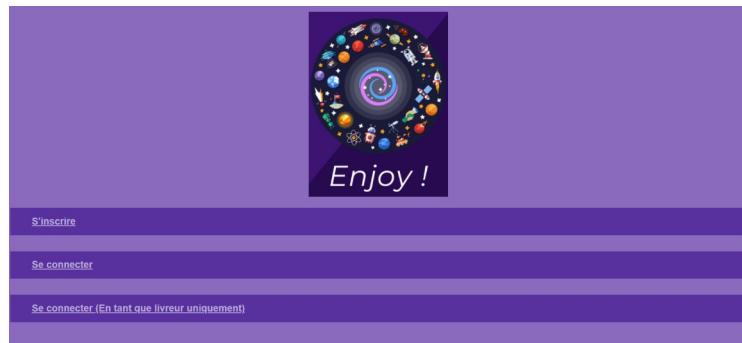
Voici nos choix généraux concernant les types des variables :

- Tous les ID sont des integer sauf ID_plat qui est un varchar 10
- Tous les noms (pays ville prénoms noms de famille) sont des varchar(25)
- Les Horaires sont de type time
- Nous n'avons pas mis de limite pour la description d'un plat en utilisant le type text
- Le code postal n'est pas un int mais un varchar (5) nous partons du principe que les restaurants sont en France

3)Description technique des fonctionnalités du site web et manuel utilisateur:

La page d'accueil :

Cette permet à l'utilisateur de choisir entre s'inscrire ,se connecter ou encore se connecter en tant que livreur .



Techniquement parlant cette page est développée uniquement en HTML/CSS et contient des liens vers les trois pages PHP.

La page d'inscription :

Permet à un nouvel utilisateur de s'inscrire en entrant ses données .

Techniquement parlant cette page contient un formulaire qui stocke les données et les insert dans la base avec une requête INSERT préparée :

```
= $cnx->prepare("INSERT INTO Client VALUES (:id_cli,:nom,:prenom,:mail,:telephone,:mdp,:num_cb,:adresse,:ville,:nb_pt_fidel);");
>bindParam(':id_cli',$new_client_id);
>bindParam(':nom',$_POST['nom']);
>bindParam(':prenom',$_POST['prenom']);
>bindParam(':mail',$_POST['mail']);
>bindParam(':telephone',$_POST['telephone']);
>bindParam(':mdp',$_POST['mot_de_passe']);
>bindParam(':num_cb',$_POST['num_cb']);
>bindParam(':adresse',$_POST['adresse']);
>bindParam(':ville',$_POST['ville']);
```

La page de connexion utilisateur :

Permet à l'utilisateur de se connecter avec son adresse mail et son mot de passe .

Techniquement parlant cette page contient un algorithme qui vas vérifier si pour cet adresse email il existe bien un client avec ce mot de passe sinon la connexion échoue .

Cet algorithme a été mis au point avec un foreach parcourant le résultat de la requête suivante :

```
$req_select = $cnx->query("SELECT mail,mot_de_passe FROM client");
```

Le foreach :

```
foreach($rows as $cle => $val){  
    if ($mail == $val['mail'] && $mdp == $val['mot_de_passe']){
        echo "<div class=\"box2\">Connecté !<br> </a></div>";
        $connexion = 0;
        $_SESSION['login'] = $mail;
        $_SESSION['mdp'] = $mdp;
        break;
    }
    else {
        $connexion = -1;
    }
}  
if ($connexion == -1){
    echo "<div class=\"box2\"> Adresse mail ou mot de passe incorrect... <br> </a></div>";
}
```

Si le client entre des données en corrélation avec un compte dans la base un message avec les liens vers le profil du client et la page de recherche de restaurant ,le client peut également choisir de se déconnecter:



La page recherche restaurants :

Cette page est la page la plus complexe du projet son rôle est de part un tri préalable paramétré par le client d'afficher les liens des pages_web des restaurants ouvert et situés dans la même ville que le client.Ainsi elle commence par demander à l'utilisateur de choisir un type de tri :

Bienvenue sur la page de recherche de nos restos !

Votre recherche

Choisissez votre type de recherche:

Par nombre d'avis

Choisissez votre type de recherche:
Par Nom du restaurant(fonctionne avec mot clefs)
Par Notes
Par nombre d'avis
Afficher tout les restos

Rechercher

Après avoir choisi un type de recherche le client clique sur rechercher et les liens vers pages des restos qui remplissent les conditions citées plus haut s'affichent :

MCDO
Burger King
O Tacos

Exemple pour un client qui habite à Meaux et qui lance la recherche à 21H47 :

id_rest	nom_rest	adresse	ville	page_web	horaire_ouverture	horaire_fermeture	prix_livraison
1	MCDO	Avenue du President Salvador Allende	Meaux	mcdo.php	08:00:00	23:59:00	5
2	KFC	2 Avenue des Sablons Bouillants	Chelles	kfc.php	11:00:00	22:00:00	7
3	Burger King	Avenue de la Haute Borne	Meaux	burger_king.php	09:00:00	23:00:00	5
4	Del Arte	Val D Europe	Chessy	del_arte.php	18:00:00	23:59:00	3
5	Pizza Hut	Avenue de la victoire	Paris	pizza_hut.php	13:00:00	23:59:00	8
6	La riviere	Boulevard de la riviere	Champs sur Marne	la_riviere.php	16:00:00	23:00:00	4
7	Le bistrot	Rue du Bistrot	Paris	le_bistrot.php	10:00:00	20:00:00	8
8	O Cafe	Allée des affaires	Pantin	o_cafe.php	15:00:00	23:00:00	3
9	O Tacos	Centre commercial aux Saisons	Meaux	o_tacos.php	18:00:00	23:59:00	5
10	Pizza 2 folies	Champs Elysees	Paris	pizza_2_folies.php	16:00:00	23:59:00	10

Les liens affichés sont bien ceux qui respectent uniquement les conditions d'ouvertures et de Localisation.

Techniquement parlant cette page renferme un algorithme qui sélectionne les pages qui sont retournés par les requêtes qui gèrent :

- la présence du restaurant dans la même ville que le client :

```
" SELECT page_web FROM restaurant,client WHERE mail = :mail AND client.ville = restaurant.ville";
```

-La vérification de l'ouverture ou non de ces restaurants :

```
SELECT nom_rest,page_web FROM restaurant WHERE CURRENT_TIME(0) BETWEEN horaire_ouverture AND horaire_fermeture AND id_rest NOT IN (SELECT id_rest FROM fermeture_exceptionnelle WHERE CURRENT_DATE IN (SELECT date_fermeture FROM fermeture_exceptionnelle))
```

-Et le tri selon le choix du client

Toutes ces requêtes sont gérées le plus efficacement possible en php afin de retourner seulement les pages filtrés avec un code propre exemple pour un tri par nom :

```
if ($_POST['type_recherche'] == "nom" && isset($_POST['recherche'])){

    $req_by_nom_sql = "SELECT page_web,nom_rest FROM restaurant WHERE nom_rest LIKE :nom";
    $req_by_nom = $cnx->prepare($req_by_nom_sql);
    $req_by_nom->execute(array(':nom'=>'%'. $_POST['recherche']. '%'));
    $rows = $req_by_nom->fetchALL();
    foreach($rows as $cle){
        foreach($opened_restos as $restos){
            foreach($client_livrable_par as $able_to_devliver){
                if(in_array($cle['page_web'],$restos)){
                    if(in_array($cle['page_web'],$able_to_devliver)){
                        echo "<a href = '".$cle['page_web']."' class=\"boutonInscription\>".$cle['nom_rest']. "</a><br>";
                    }
                }
            }
        }
    }
}
```

La page d'un restaurant :

Après avoir choisi un restaurant parmi ceux retourné par notre algorithme le client peut cliquer sur celui-ci et il sera renvoyé vers la page web de ce restaurant.

Sur cette page il peut consulter les plats présentés par ce restaurant ainsi que leurs prix .

Le client peut également concocter une nouvelle commande avec les plats qu'il souhaite avec la quantité qu'il souhaite , le prix de la commande s'actualise au fur et à mesure que le client ajoute des plats .

Lorsque le client clique sur commander la commande est envoyé à notre plateforme et est gérée par nos livreurs situés dans la même ville que le client et le restaurant .

Ainsi si aucun livreur n'est disponible il sera impossible pour le client de passer une commande (il y aura un message de prévention à la place)

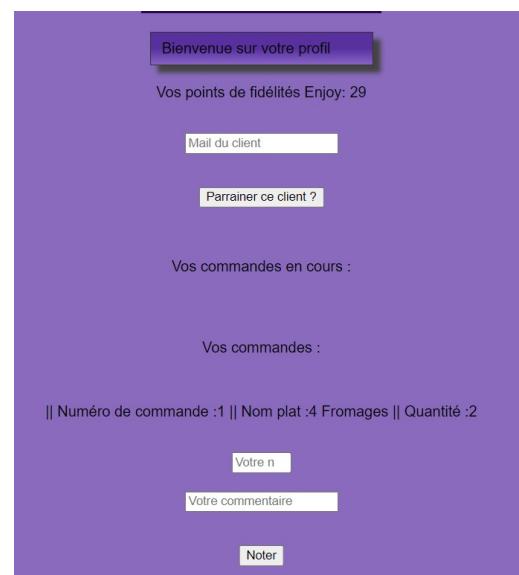
Si un livreur est disponible alors elle sera automatiquement prise en charge et son statut passera à En Livraison dans la base de données un message de confirmation est également affiché au client.

Techniquement parlant les pages restaurants sont plutot simple mais les requêtes qui les composent son variés UPDATE,SELECT,INSERT

La page de profil personnel :

Cette page permet au client de :

- voir ses points de fidélités (+50 par commande et par parrainage)
 - de parrainer un client en renseignant son adresse mail (le compte du parrain sera lié à celui du client parrainé dans la base de données)
 - de consulter ses anciennes commandes
 - de noter ses anciennes commandes et de les commenter (le commentaire et la note sont enregistrés dans la base de données)
- Egalement basique techniquement parlant .



La page de connexion pour livreur :

Cette page permet à un livreur de se connecter avec son matricule et son mot de passe. Techniquement parlant c'est exactement la même que celle de connexion utilisateur mais elle vérifie le matricule au lieu de l'adresse mail en plus du mot de passe ainsi ses requêtes pointent sur la table livreur et non la table client.

Une fois le livreur identifié il peut ensuite consulter son profil professionnel à l'aide d'un lien qui le redirige vers celui-ci :

Livreur ? Connectez vous

Votre matricule

Votre mot de passe

Se connecter

Connecté !

[Votre profil](#)

La page de profil professionnel :

Cette page permet au livreur de choisir la ville où il souhaite travailler et choisir son statut entre en ligne,hors ligne en attente de commande et en livraison.

Ces choix sont évidemment enregistrés dans la base et sont liés à la possibilité ou non de commander dans certaines ville (pas de livreur disponible impossible de commander).

Initialement le livreur pouvait choisir de prendre en charge une commande mais par soucis technique nous avons décider que cela allait se faire automatiquement . (pour voir le code qui s'en occupe aller dans les pages de restaurants lignes 73-94).

Techniquement parlant cette page est très simples et contient des requêtes basiques.

En Ligne

Changer mon statut ?

Votre statut :En Ligne

Listes commandes

Ville

Couvrir cette ville ?

4)Pistes d'améliorations de notre travail :

Au niveau des pistes d'amélioration de notre travail :

-Ajout de contraintes dans nos tables sql pour éviter deux clients avec les mêmes données par exemple.

-Ajout d'un système « mot de passe oublié ? » qui permet de changer son mot de passe après un oubli, on vérifiera que l'utilisateur qui veut changer le mot de passe est bien le propriétaire du compte en lui demandant d'entrer son numéro de téléphone adresse et code cb.

-Ajout d'une fonction « retour à votre profil » par exemple que l'utilisateur cherche un restaurant il est obligé de se reconnecter si il veut regarder son profil .

5)Organisation et répartition du travail :

Afin de mener à bien ce projet , nous avons fait un groupe discord afin de faciliter les envois et la communication concernant le projet . Au niveau de la répartitions du travail , Nelson s'est occupé de la partie PHP/SQL et graphique , Mikael s'est occupé de la partie SQL et enfin Pierre a fait un peu de tout (partie SQL et php/graphique) .