

Vue globale de notre Projet :

Sujet numéro 7 : Application Web à usage interne de l'entreprise : du full-covoiturage domicile-travail, domicile-loisir et travail-loisir.

- Dans ce document répertoriant la vue globale de notre projet, notre **organisation** et **méthodologie** de travail y est représenté.
 - Afin de réaliser un travail professionnel, nous avons dû passer par la réalisation de :
 - Document .txt comportant notre reformulation du cahier des charges ;
 - Croquis et nombreux dessins sur tableau ;
 - Dessins d'interfaces de notre site ;
 - Schémas d'interfaces utilisateurs et serveur ;
 - De nombreux autres documents qui seront également développés dans ce document.
-
- Dans ce type de projet, le point de départ est important afin de ne pas s'éparpiller et surtout garder en vue la réalisation finale. C'est pour cela que d'après les conseils que l'on nous a donné, nous avons pris la décisions de faire une courte reformulation du cahier des charges. Cette dernière n'est pas complète, mais elle comprend les principaux points que nous devons mettre en place dans notre application web. La voici :

*****reformulation cahier des charges** :***

- outils destiné aux étudiants de notre formation.
- mise en commun des étudiants qui sont dans la même ville, qui vont à un endroit pour les études, les loisirs ou les courses.
- application va proposer aux étudiants de faire un trajet ensemble s'ils ont les mêmes horaires.
- pouvoir faire ça selon la géolocalisation (cartographie) de ce que fait l'étudiant comme énoncé en haut.
- mettre différents moyens de transports : pieds, vélo, bus, voiture personnelle.
- si utilisation d'une voiture plusieurs choses doivent être renseignées en plus comme : véhicule, nombre de places disponibles, participation demandée (financière ou autre).
- créer une application si temps qui permet de voir ça pour tous les jours de la semaine.
- les étudiants doivent renseigner où ils habitent, s'ils utilisent un véhicule (type, nb places, immatriculation, etc., carte grise, contrôle technique, assurance), où il fait les courses et loisirs (adresse, arrêt de bus, moyen de transport utilisé, durée moyenne).
- des choses peuvent être rassemblés dans la base de données comme l'arrêt de bus qui est le lieu de départ, durée moyenne avec des calculs...

- **pas compris** : nous renseignerons ses informations universitaires : formation, groupe ; sous-groupe, le début de cours (-15') et la fin de cours (+15') pour chaque jour de la semaine à venir (lundi au samedi).

- il faut utiliser des fichiers spécifiques comme .json ou csv pour remplir la base de données.

- on peut remplir la base depuis notre site directement ou sur phpmyadmin.

- il faut détecter qui est proche au niveau des lieux pour les mettre ensemble selon les demandes et proposer différents choix (les personnes qui proposent proche, pas que un seul choix).

- **Pas compris** : Chaque participant de l'équipage aura à valider ou invalider un trajet aller et un trajet retour par jour de la semaine.

- Permettre à une personne de changer de covoiturage et d'en prendre un autre (changement en cascade dans la Base de Données).

- possibilité d'ajouter une nouvelle voiture (un étudiant peut avoir plusieurs voitures tout comme ne pas en avoir), il peut aussi supprimer son véhicule.

- interface de gestion des informations pour organiser des déplacements en groupe, voire un co-voiturage des étudiants.

=> ne pas oublier de réaliser un court tuto pour montrer comme notre site fonctionne.

Cette reformulation du cahier des charges nous auras aussi permis de poser des questions sur des critères que nous n'avions pas compris.

- Afin d'avoir une base de données complète, nous avons réunis les éléments qui fallait y mettre et nous avons alors pu commencer le Modèle Conceptuel de base de Données comme vu dans la ressources « source de données » :

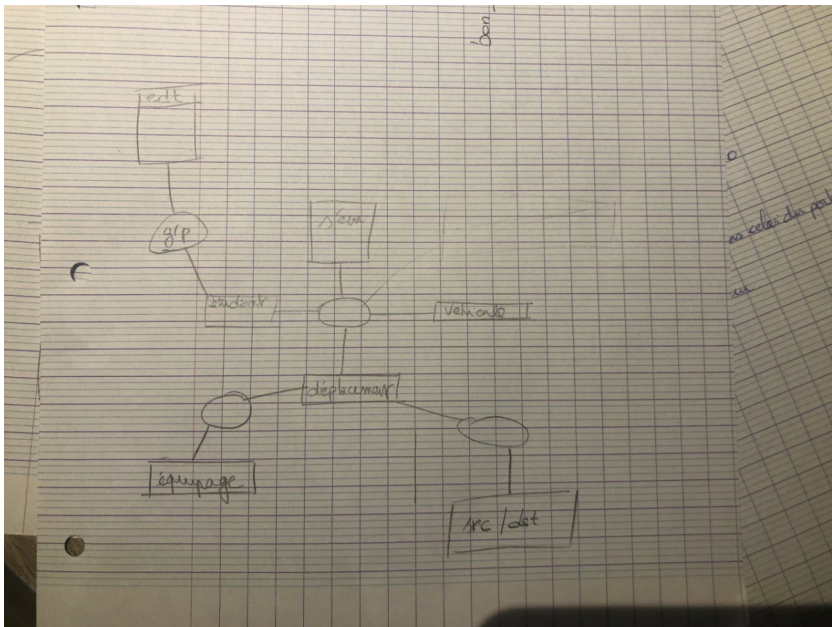


Figure 1: Première version du MCD sur feuille répertoriant les principales cardinalités

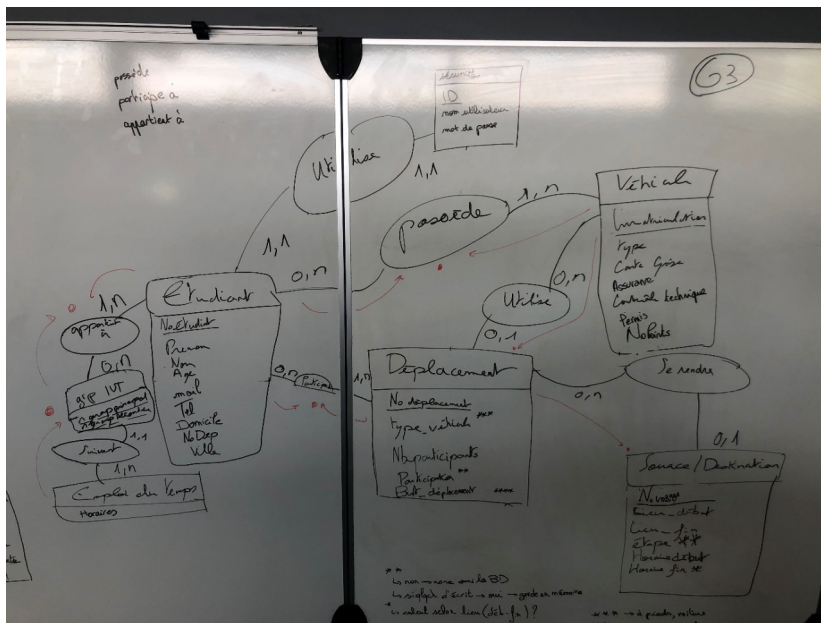


Figure 2: Schéma du MCD final réalisé au tableau de notre salle

- Ce dernier MCD nous a permis la réalisation du MPD, qui aura eu une assez grande amélioration entre les 3 versions. En effet, nous sommes passé par une simplification du MPD afin d'obtenir une base plus simple à utiliser et surtout plus cohérente que l'ancienne :

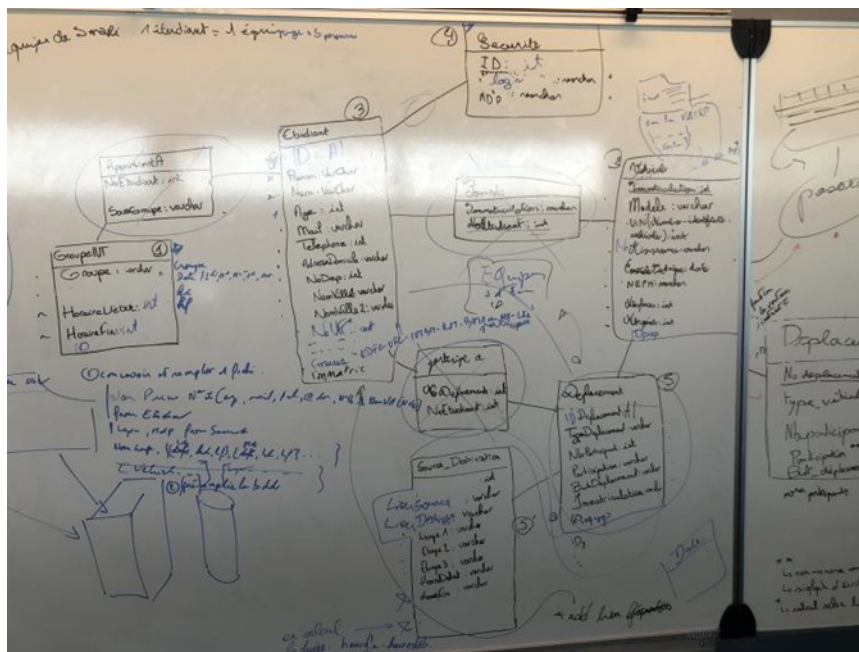


Figure 3: Première version de notre MPD

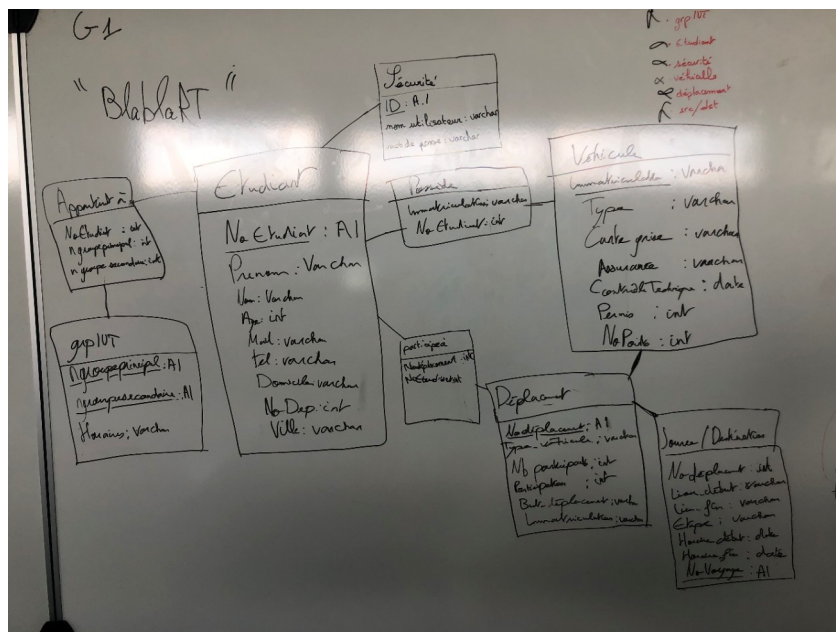


Figure 4: Version presque finale de notre MPD

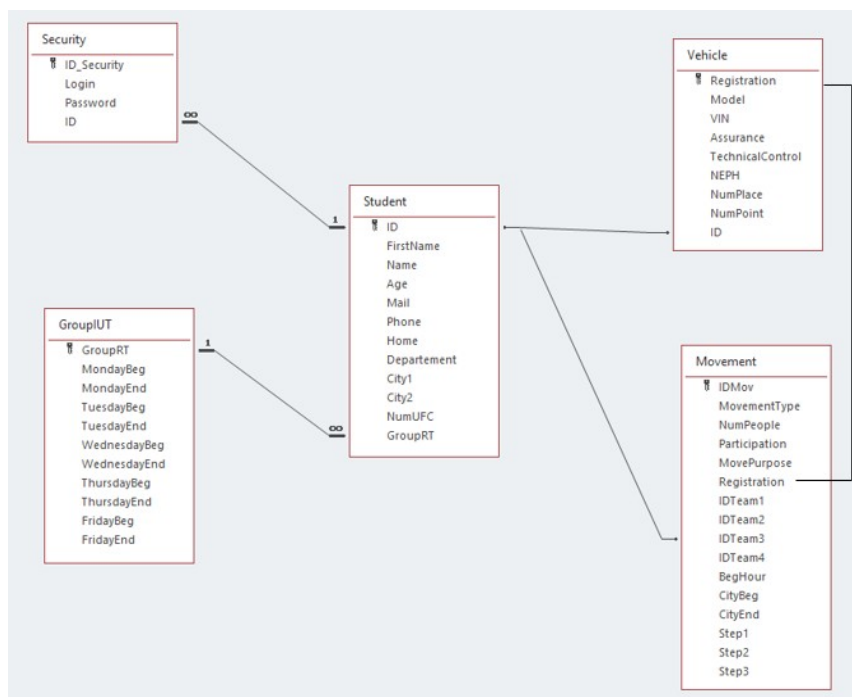


Figure 5: Version finale de notre MPD sur Accès

- Nous avons pu ainsi commencer les schémas du visuel de notre site web. Le premier schéma ci-dessous nommé Figure 6, est notre premier brouillon. Le visuel que nous avons proposé à ce moment n'a pas vraiment été retenu. Puis nous avons décidé d'en refaire un plus propre, qui propose toutes les interfaces aux utilisateurs selon le choix de ce dernier (Figure 7).

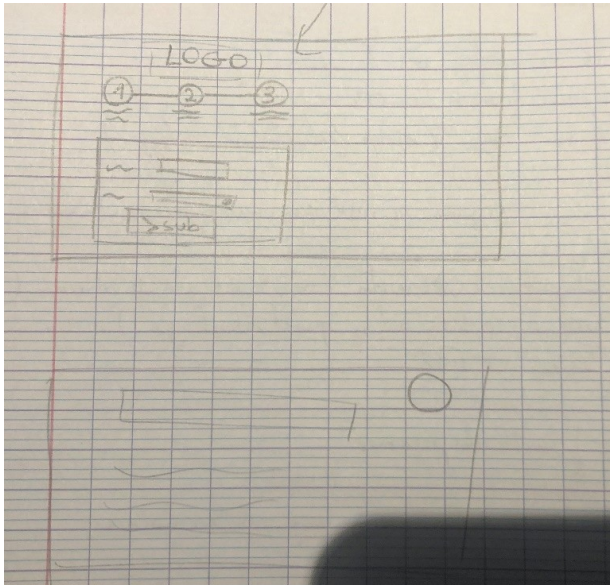


Figure 6: Premiers visuels de notre application web

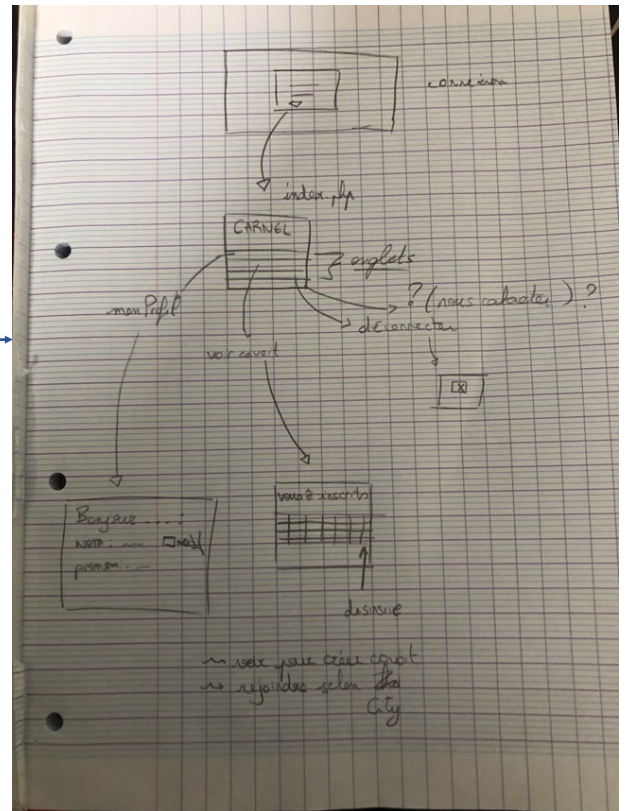


Figure 7: Visuels finaux de notre application web

- Lors de l'utilisation de la base de données, afin de mieux voir les liaisons et l'impacte des clés, nous avons décidé de réaliser un « mini » modèle physique représentant l'impact et le sens des clés :

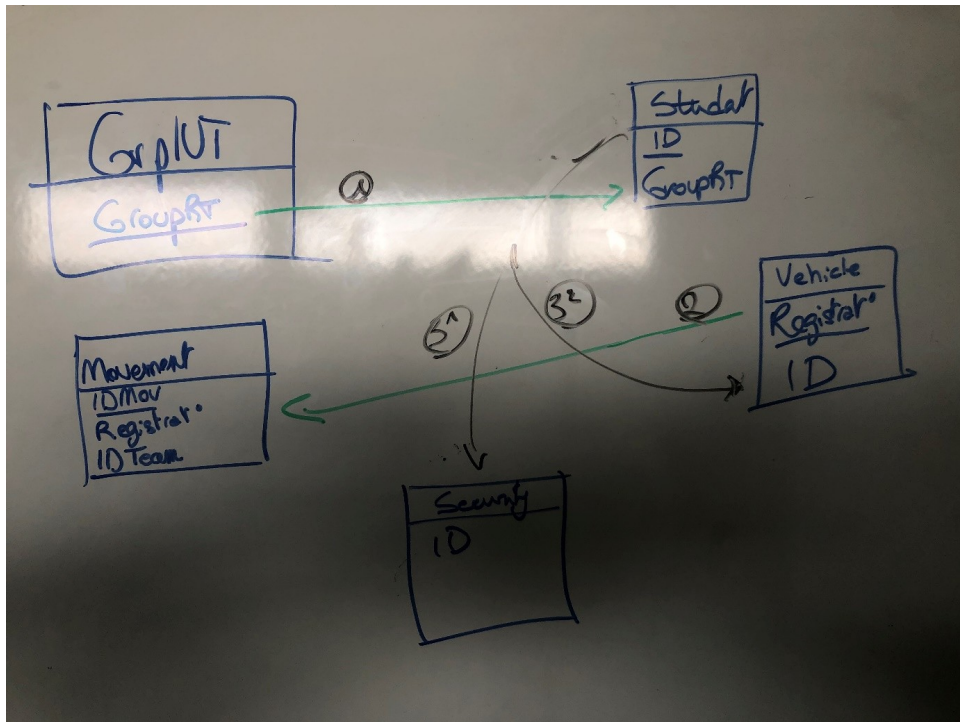


Figure 8: Petit schéma des liens avec les clés entre les bases de données

- Nous avons aussi réalisé par la suite des schémas de gestion des interfaces entre « end user » et côté serveur-base de données :

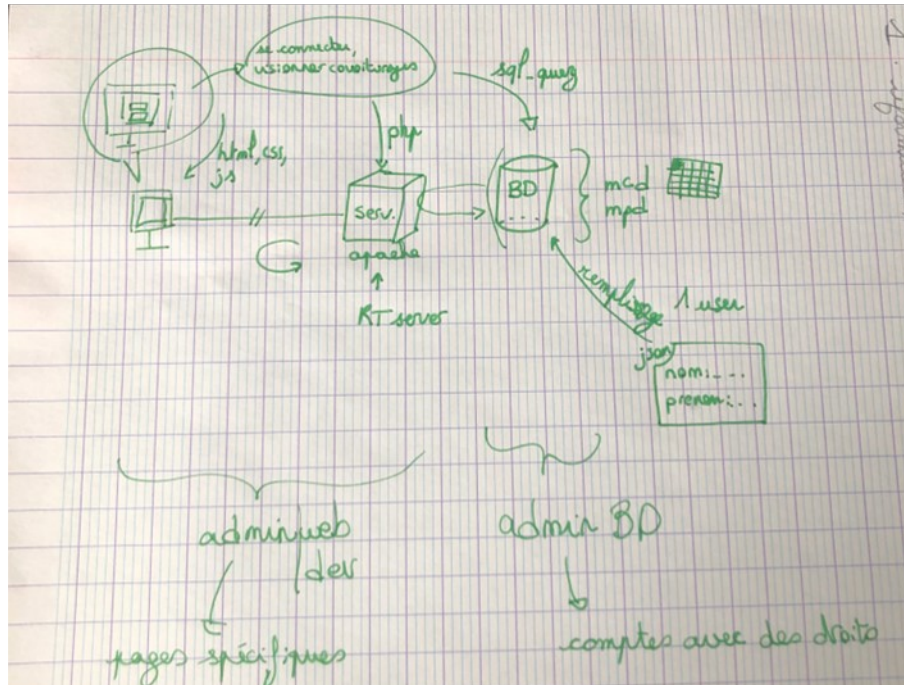


Figure 9: Premières version du schéma de gestion des interfaces

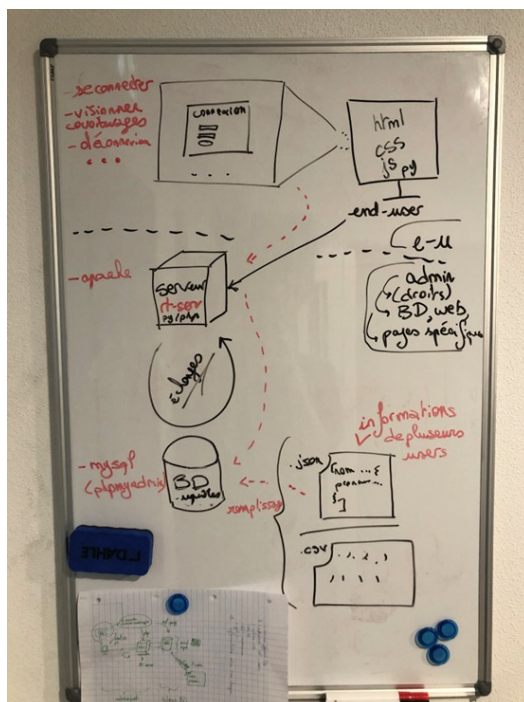


Figure 10: Version finale du schéma de gestion des interfaces

- Nous avons aussi décidé de réaliser un logo pour notre application web se nommant Cernel. Venant des voitures « car » et du kernel (noyau du système d'exploitation Linux).



Figure 11: Logo de Cernel

- Nous avons utilisé plusieurs technologies lors du développement de notre application web :
 - S'exécutant côté client :

➔ HTML : architecture de la page web :

Entrez votre nom d'utilisateur

Entrez votre mot de passe

Figure 12: Page de connexion uniquement en html

➔ CSS : travail sur le côté visuel de la page :

La page de connexion est présentée dans un cadre orange. Au centre, il y a un formulaire blanc avec des bordures arrondies et une ombre portée. Le formulaire contient le logo Cernel en haut à gauche, deux liens "Connexion" et "Voir Identifiants" en orange, des champs de saisie pour "Nom d'utilisateur" et "Mot de Passe" (avec un bouton pour masquer/montrez le mot de passe), et un bouton orange "Connexion" en bas.

Figure 13: Page de connexion avec le CSS en plus

➔ **JavaScript** : création du boutons pour voir le mot de passe.

- S'exécutant côté **serveur** :

➔ **PHP** : lier le site à la base de données

```
////////////////////////////////////////
$informations = substr($informations,0,-2);
$informationskey = substr($informationskey,0,-2);
//echo "<br>" . $informations;
//echo "<br>" . $informationskey;
$sql1= "INSERT INTO Student ($informationskey) VALUES ($informations)";
//echo "<br>" . "cbr>".$sql1;
if (mysqli_multi_query($conn, $sql1)) {
    echo "New records created successfully";
} else {
    echo "Error: " . $sql1 . "<br>" . mysqli_error($conn);
}
```

Figure 14: Exemple de requête SQL dans un code en PHP

➔ **Python** : rentrer le fichier JSON dans la base de données :

```
61 assurance=data[2]["Vehicule"]['Assurance']
62 controletechnique=data[2]["Vehicule"]['ControleTechnique']
63 NEPH=data[2]["Vehicule"]['NEPH']
64 noplaces=data[2]["Vehicule"]['NoPlaces']
65 nopoints=data[2]["Vehicule"]['NoPoints']
66
67 #-----
68
69 typedeplacement=data[3]["Deplacement"]['TypeDeplacement']
70 nbrparticipants=data[3]["Deplacement"]['NbrParticipants']
71 participation=data[3]["Deplacement"]['Participation']
72 butdeplacement=data[3]["Deplacement"]['ButDeplacement']
73
```

Figure 15: Code en Python utilisé pour rentrer dans la base de données les informations du fichier JSON

➔ Fichiers JSON : fichier contenant les informations que nous souhaitons rentrer dans la base de données :

```
1  [
2  {
3    "Informations":
4    {
5      "Prenom": "Thomas",
6      "Nom": "Raynaud",
7      "Age": 19,
8      "Mail": "47.4667",
9      "Telephone": "47.4667",
10     "Adresses":
11     {
12       "AdresseDomicile": "5 rue de la piscine",
13       "NoDep": "68250",
14       "NomVille": "Rouffach",
15       "NomVille2": "Montbéliard"
16     }
17   }
18 },
```

Figure 16: Partie fichier JSON contenant les informations d'un utilisateur.

➔ **Requêtes SQL** : Remplir et demander des informations dans la base de données (en PHP, voir Figure 14).

➤ Matériel (**Hardware**) et logiciels (**Software**) utilisés :

- D'un point de vu **hardware** nous avons utilisé nos ordinateurs portable ainsi que les ordinateurs présents à l'IUT afin de réaliser le code.
- Pour le **software**, nous avons utilisé plusieurs éditeurs de texte comme Atom ou Visual Studio Code.
Nous avons utilisé Microsoft Acces afin de simuler notre base de données.
Nous avons aussi utilisé des serveurs apache et une base de données phpmyadmin sous XAMPP afin d'exécuter le code en PHP et d'utiliser une base de données comme ce que nous avons faire à l'aide de FileZilla sur le serveur rt-projet. Pour les rendus, nous avons utilisé Word, Power Point ainsi que github pour déposer notre code bien qu'il ne puisse s'y exécuter à cause du manque de serveur PHP.
Nous avons réalisé la base de données phpmyadmin en anglais afin de faciliter le travail à un administrateur non francophone.

- Il faut savoir que les utilisateurs dans notre sites ont des comptes pré-crées grâce au fichier JSON. Ces derniers ont alors accès seulement aux pages qui leurs sont réservés (connexion, voir leurs covoiturage, voir tous les covoiturages avec des filtres, déconnexion). Tandis que des administrateurs ont plus de choix, ils ont notamment à la place de la page « voir mes covoiturages » une page qui les contiens tous, ils peuvent alors décider de les supprimer si besoin.

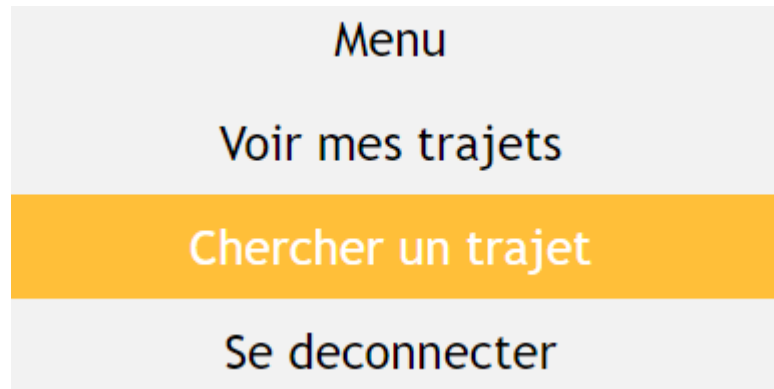


Figure 17: Onglet de navigation entre les pages de notre site web.

Plusieurs types d'administrateurs existent, tel que administrateur de base de données avec le choix de créer des lignes, ajouter des données (manuellement ou avec le script php pour entrer le fichier JSON), ou même supprimer des tables. L'autre type d'administrateur serait un administrateur web, qui a accès aux page réservées avec un compte spécifique énoncé précédemment ce dernier pourrait aussi modifier les codes réalisés.

- Les principales fonctionnalités de notre application du côté utilisateur :
 1. Se connecter à l'aide d'un login et d'un mot de passe. L'utilisateur peut vérifier s'il a fait une erreur dans le mot de passe à l'aide d'un bouton qui cache et montre ce dernier.
 2. Une page menu. Présentation du site et du contexte de création de ce dernier. Menu présentant les différentes pages utilisables.
 3. « Voir mes trajets », visionner les trajet de l'utilisateur connecté. Plusieurs informations sont disponibles, comme le but du déplacement, le nombre de personnes, qui sont les personnes présentes, un moyen de se désinscrire du déplacement désigné. Nous n'avions pas eu le temps d'ajouter quand à lieu le covoiturage bien que ce n'est pas compliqué en vu de la facilité de notre base de données.
 4. « Chercher un trajet », contient un formulaire qui initialement montre tous les covoiturages, mais nous pouvons affiner les recherches à l'aide de filtre (ville de départ, ville d'arrivé, étape1, étape2, étape3, heure et minute de départ). Nous avons prévu d'ajouter un bouton permettant de rejoindre et créer un covoiturage mais nous n'avions pas eu le temps de finir le code.

5. «Se déconnecter », boutons renvoyant vers la page de connexion et qui ferme la session en cours d'utilisation.
6. Si nous sommes connecté à un compte administrateur, nous avons le moyen de voir tous les trajets et d'en supprimer à la page « voir mes trajets ».