



# RAPPORT PERSONNEL DE PROJET CAPS CORENTIN

18.01.2021

---

<b>Rappel de ma partie</b>	<b>2</b>
<b>Base de données</b>	<b>2</b>
<b>Développement du site web</b>	<b>6</b>
Codage	6
Css	12
<b>IV. Résultat final</b>	<b>16</b>
<b>V. La boucle de courant</b>	<b>18</b>
<b>VI. Conclusion</b>	<b>19</b>

## I. Rappel de ma partie

Ma partie consiste dans un premier temps à développer une base de données pour stocker les informations du parking.

Dans un second temps à développer un site web permettant de consulter en temps réel le taux d'occupation du parking

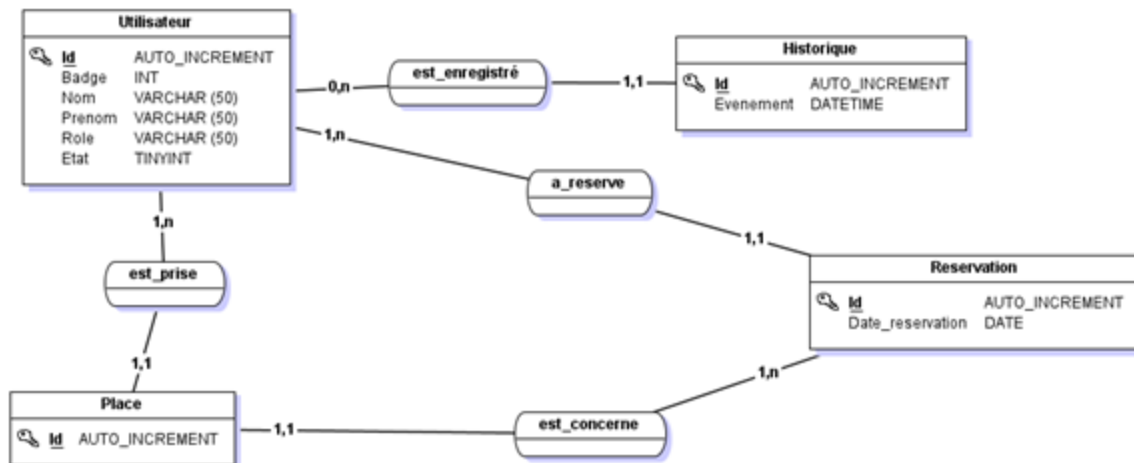
## II. Base de données

Pour créer la base de données j'ai tout d'abord réfléchi à ce que devait contenir cette base. J'en ai conclu qu'elle devait contenir, 4 tables :

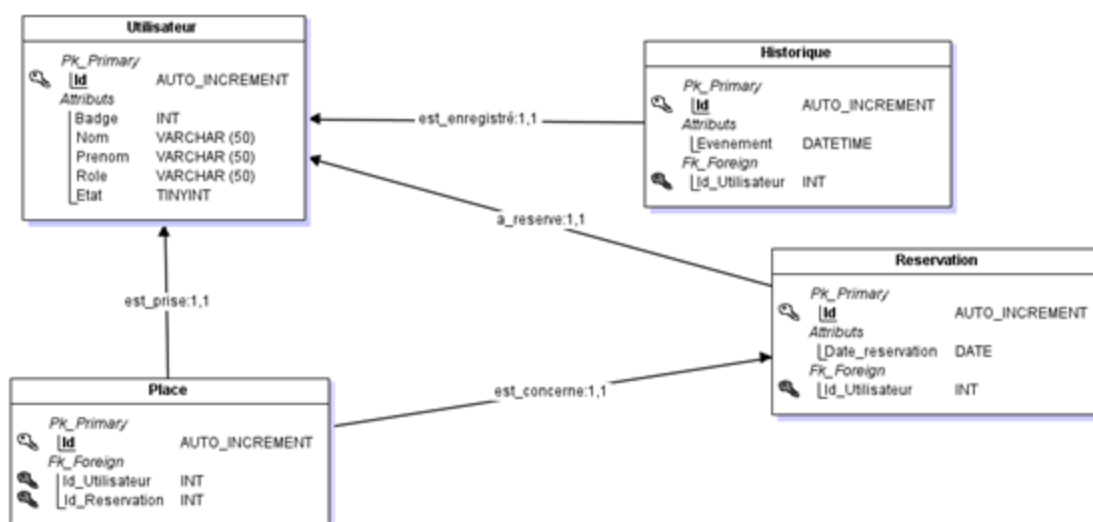
- Une table utilisateur avec 6 champs;
  - o id est la clé primaire.
  - o id badge est une chaîne de caractère qui correspond au numéro de badge de l'établissement.
  - o nom qui est une chaîne de caractère qui correspond au nom et prénom de la personne.
  - o prénom est une chaîne de caractère également et qui correspond au prénom du détenteur du badge.
  - o rôle qui est aussi une chaîne de caractère et qui correspond au rôle de l'utilisateur par rapport au parking (simple utilisateur ou administrateur).
  - o état qui est un booléen pour savoir si l'utilisateur peut accéder au parking ou s'il est suspendu.

- Une table historique qui permet de garder une trace de l'activité du parking. Cette table contient un champ id qui est la clé primaire de la table, un champ événement qui est du type datetime qui permet de connaître l'heure et la date d'ouverture de la barrière du parking et un champ id\_Utilisateur qui est une clé étrangère vers la table utilisateur et qui permet d'associer l'id d'un utilisateur à une ouverture de la barrière.
- Une table réservation qui permet d'enregistrer dans la base de données la réservation d'une place par un utilisateur. Elle contient 3 champs; un champ id qui est la clé primaire de la table, un champ date réservation qui est de type date et qui permet d'enregistrer le jour de la réservation, et un champ id\_utilisateur qui est une clé étrangère vers la table utilisateur et qui associe l'id d'un utilisateur à une réservation.
- Une table place qui contient 4 champs ; un champ id qui est la clé primaire de la table, un champ état qui est de type enum (libre ou occupé) et qui permet de savoir si une place est disponible ou indisponible, un champ numéro qui correspond au numéro de la place dans le parking, et un champ id\_capteur qui correspond à l'identifiant du capteur associé à cette place

Pour créer cette base je suis passé par plusieurs étapes. Tout d'abord, j'ai utilisé le logiciel jmerise pour créer un schéma entité-association de la base de données.



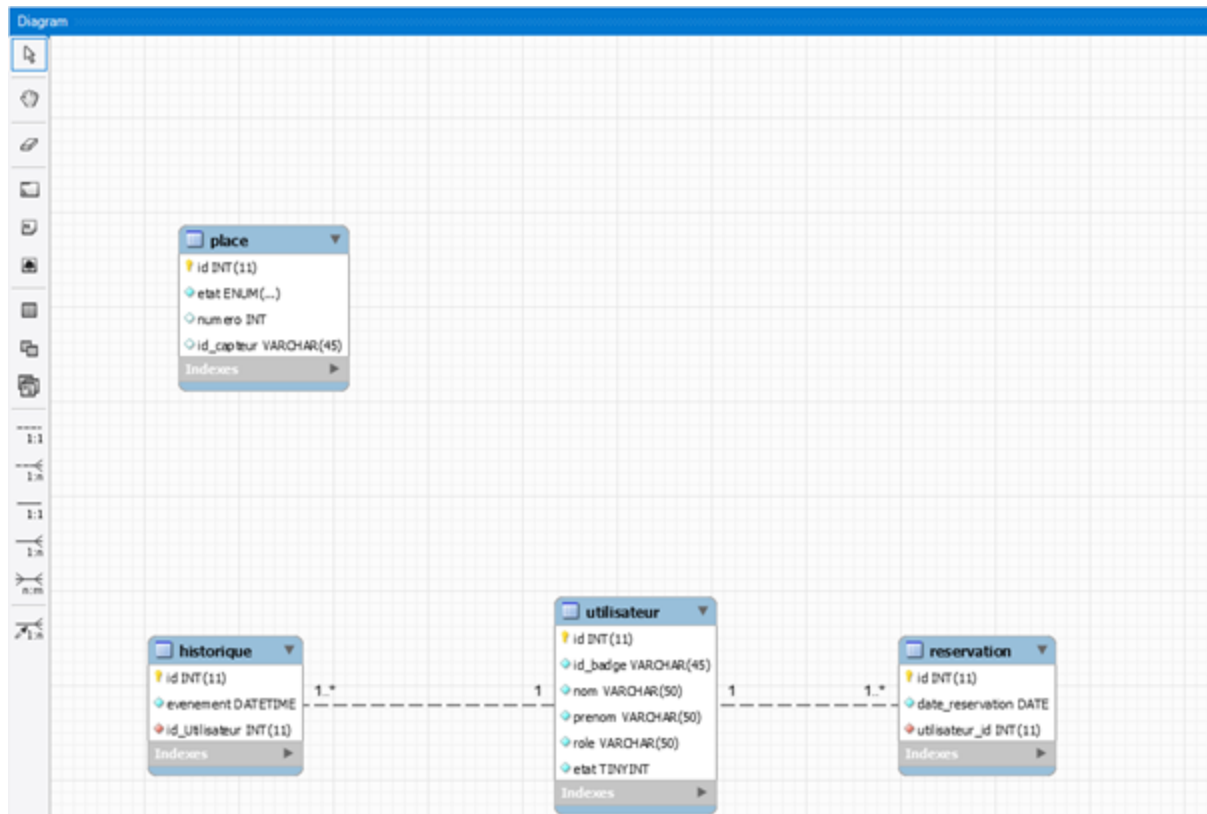
J'ai ensuite utilisé le même logiciel pour concevoir un schéma relationnel à partir du modèle entité association précédemment créé.



Enfin, à partir de ce dernier schéma le logiciel JMerise a généré un script SQL que j'ai pu enregistrer et utiliser pour créer la base de données sur Wamp Serveur. Les différents schémas étant créés nous avons commencé à travailler avec cette première base de données. Il s'est avéré qu'elle n'était pas correcte car j'avais fait des erreurs lors de la conception. J'ai donc décidé de reprendre le schéma relationnel, mais avec le logiciel MySQL Workbench cette fois, car c'est un logiciel que je maîtrise mieux. J'ai aussi fait des modifications directement dans la base de données grâce à la commande SQL ALTER TABLE qui est la commande permettant entre autres d'ajouter, de supprimer ou de modifier un champ.

Mais ces modifications ne sont pas portées sur toute la base, la seule table à avoir subi de réelle modification est la table place, dans laquelle j'ai rajouté le champ état, le champ numéro et le champ id\_capteur. J'ai supprimé les champs id\_utilisateur et id\_réservation qui étaient des clés étrangères vers les tables utilisateur et réservation.

Enfin, après avoir fait toutes les modifications pour que la base de données soit fonctionnelle et la plus optimale possible, j'ai utilisé la fonction reverse engineer de MySQL Workbench qui est une fonction qui permet de créer un schéma relationnel à partir d'une base de données déjà existante. J'ai obtenu ce schéma relationnel.



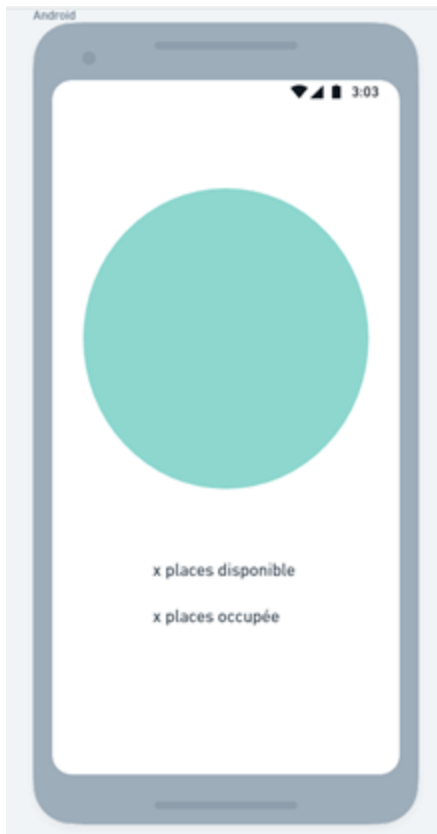
Enfin pour permettre un accès plus simple à la base de données par tous les membres du groupe, nous avons transféré la base de données sur le serveur Web de la section. Elle est désormais accessible par l'adresse 10.6.0.1:8044. J'ai également créé un utilisateur avec le nom parking et le mot de passe Nantes44.

### III. Développement du site web

#### a) Codage

La deuxième tâche que j'ai dû réaliser est le développement d'un site web permettant la visualisation de l'occupation du parking en temps réel; pour ce faire j'ai utilisé l'IDE apache netbeans.

Avant de commencer le développement, j'ai réalisé une maquette de ce que pourrait être le site final pour m'aider dans ce développement. J'ai obtenu cette maquette.





Une fois la maquette réalisée, j'ai pu commencer le développement du site web. La première étape fut de connecter le site web à la base de données que j'avais créé précédemment. Pour ce faire j'ai renseigné les informations de la base de données en arguments de la fonction PARAM ces arguments sont :

- \$PARAM\_hote qui est le chemin vers le serveur contenant la base de données ce chemin est 10.6.0.1
- \$PARAM\_port qui est le port utilisé 3306.
- \$PARAM\_nom\_bdd qui est le nom de la bdd dans le serveur notre bdd s'appelle parking.
- \$PARAM\_utilisateur qui est le nom de l'utilisateur de la bdd il s'appelle parking également.
- \$PARAM\_mot\_passe qui est le mot de passe de l'utilisateur pour se connecter à la base de données ce mots de passe est Nantes44

Ces arguments sont ensuite utilisés dans la fonction connexion pour créer un nouvel objet de connexion à MySQL et faire le lien avec la base de données.

```
$PARAM_hote='10.6.0.1'; // le chemin vers le serveur
$PARAM_port='3306';
$PARAM_nom_bd='parking'; // le nom de votre base de données
$PARAM_utilisateur='parking'; // nom d'utilisateur pour se connecter
$PARAM_mot_passe='Nantes44'; // mot de passe de l'utilisateur pour se connecter
$connexion = new PDO('mysql:host='.$PARAM_hote.';port='.$PARAM_port.';dbname='.$PARAM_nom_bd, $PARAM_utilisateur, $PARAM_mot_passe);
```

Une fois la connexion effectuée, j'ouvre une requête SQL grâce à la fonction `$resultats` qui utilise la fonction `$connexion` en ouvrant une requête. En l'occurrence la requête que j'utilise ici est `SELECT* FROM place` cette requête va sélectionner tous les éléments de la table `place`, une fois cette requête effectuée on utilise à nouveau la fonction `$resultats` mais cette fois avec la méthode `setFetchMode` qui permet de définir le mode de récupération de la requête ici on la définit sur un objet.

```
<?php
$resultats = $connexion->query("SELECT * FROM place ");
$resultats->setFetchMode(PDO::FETCH_OBJ);
```

Une fois la requête effectuée, j'ai voulu afficher la date et l'heure pour que les personnes utilisant le site web puissent consulter les disponibilités du parking en temps réel. J'ai utilisé dans un premier temps uniquement la fonction `$date` et `$heure`, mais cela ne m'a pas donné la bonne heure, j'avais un décalage de deux heures. J'ai donc associé aux fonctions utilisées précédemment le `date_default_timezone_set` qui me permet de définir par défaut le fuseau horaire que j'utilise et d'avoir l'heure désirée.

```
date_default_timezone_set("Europe/Paris");
$date = date("d/m/Y");
$heure = date("H:i");
```

Une fois la date et l'heure paramétrée, j'ai commencé à développer la solution qui allait permettre d'afficher les places du parking utilisées ou non. Pour cela j'ai commencé par initialiser une fonction à 0, cette fonction `$a` permettra de stocker le nombre de places du parking dont l'état sera celui qu'on a défini dans la boucle.

Une fois cette fonction initialisée, j'ai créé une boucle `for` avec comme paramètre `$i = 0`, `$i <= 5`, `$i ++` ces paramètres signifient que la boucle part de 0 va jusqu'à 5 maximum et s'incrémente de 1 à chaque passage.

Cette boucle sera connectée à la base de données, car elle a besoin d'avoir accès à l'état de la place et à son id pour pouvoir les compter. Cette connexion se fera grâce aux fonctions `$place` qui appelle `$resultats` et ensuite on réutilise `$place` pour accéder à l'id et à l'état de la place.

La boucle for ayant accès à l'état des places, j'ai créé une boucle if qui incrémente \$a de 1 lorsque l'état d'une place dans la base de données est 'occupé'.

```
$a = 0;

for($i = 0; $i <=5; $i ++){
    $place = $resultats->fetch();
    $id = $place->id;
    $etat = $place->etat;

    if($etat === "occupe")
    {
        $a = $a + 1;
    }
}
```

Une fois que toutes les places 'occupé' ont été stockées dans la fonction \$a, j'ai créé une fonction \$diff qui permet de faire la différence entre \$a et le nombre total de places dans le parking. J'ai ensuite créé une fonction if qui lorsque \$diff est supérieure à 0 affiche le nombre de place restante (le résultat de \$diff), et qui affiche également \$date et \$heure, cette fonction if affiche également un rond vert s'il y a des places disponibles dans le parking.

```
$diff = 6 - $a;

if($diff > 0)
{
    Place disponible : ' . $diff . '</br> Nous sommes le : ' . $date . ' </br> il est : ' . $heure . ' </font> </p>'

    <div class="position">
    
    </div>
}
```

J'ai ensuite fait un else, dans le cas où \$diff est égale à 0, et donc que le parking est complet. Dans ce cas l'affichage est presque identique ; les seules différences sont que l'on affiche plus le nombre de places mais "parking complet" et le rond vert passe au rouge.

```
Parking plein </br> Nous sommes le : ' . $date . ' </br> il est : ' . $heure . ' </font> </p>';

<div class="position">

</div>
<?php
```

Afin que l'utilisateur n'ai pas besoin d'actualiser manuellement la page pour connaître les disponibilités du parking, j'ai créé un script permettant une actualisation automatique toutes les 10 secondes. Le temps de recharge est défini par le `setInterval` (en milliseconde) et ensuite on appelle la fonction `reload` qui permet de rafraichir la page

```
<script type="text/javascript">
  setInterval("reload();",10000);

  function reload(){
    window.location = location.href;
  }
</script>
```

## b) Css

Une fois le codage du site terminé, j'ai voulu que le site soit responsive c'est à dire qu'il fonctionne aussi bien sur ordinateur, sur tablette ou sur téléphone pour cela j'ai utilisé @media pour que lorsque la taille de l'écran passe sous les 780 pixel la taille de la police diminue pour s'adapter à l'écran.

```
@media (max-width: 780px)
{
    p{
        font-size: 30px !important;
    }
}
```

Une fois les tailles d'écran réglées, j'ai défini les réglages du body, j'ai réglé les marges sur 0 et la taille sur 100% pour que la page web utilise toute la page. J'ai défini la police sur Montserrat sans sérif la couleur du fond en blanc (#ffffff en hexadécimale). J'ai également défini une image de fond sur le site web, j'ai utilisé la fonction background-image qui permet d'accéder à une image grâce au chemin de cette dernière. J'ai aussi utilisé la fonction background-repeat : no-repeat pour que le fond soit une image unique, et pas la répétition de plusieurs fois l'image en plus petit. J'ai défini la taille que devait faire l'image pour remplir la page à savoir 1950 pixels par 1000 pixels.

```
body{
    margin: 0px;
    padding: 0px;
    width: 100%;
    font-family: 'Montserrat', sans-serif;
    background-color: #ffffff;
    background-image: url(image/images.jpg);
    background-repeat: no-repeat;
    background-size: 1950px 1000px;
}
```



Une fois le bandeau terminé j'ai défini les paramètres de l'image servant à indiquer si le parking est plein ou s'il y a des places disponibles. Pour qu'elle soit centrée j'ai utilisé la fonction transform : translate (-50%) et position absolue pour qu'elle soit toujours au centre de l'écran j'ai aussi défini le z-index sur 1 pour que l'image soit par-dessus l'image de fond.

```
.position img
{
    left: 50%;
    transform: translate(-50%);
    position: absolute;
    z-index: 1;
    top: 15%
}
```

Pour finir j'ai créé une balise que j'ai utilisé pour le texte sur le reste de la page. Dans cette balise j'ai réutilisé la fonction transform : translate (-50%) pour disposer le texte au centre de l'image, et j'ai aussi utilisé la fonction position : absolute. J'ai utilisé text-align : center pour que le texte soit centré et non aligné à gauche. J'ai défini la taille de police sur 50 pixels pour qu'elle reste lisible sans prendre trop de place, et j'ai défini le z-index du texte sur 2 pour qu'il soit par-dessus l'image de fond et l'image du rond.

```
p
{
    left: 50%;
    transform: translate(-50%);
    position: absolute;
    text-align: center;
    font-size: 50px;
    face: 'verdana';
    color: white;
    bottom: 29%;
    z-index: 2
}
```



Ces deux balises sont utilisées dans le main dans la boucle if permettant d'afficher le taux d'occupation du parking

```
<div class="position">  
  
</div>
```

```
'<p> Place disponible : '
```

## IV. Résultat final

Lorsque tout est assemblé, j'ai simulé l'occupation du parking en insérant des places libres et occupées manuellement dans la base de données, ce qui donne lorsqu'il y a des places disponibles dans le parking le visuel suivant :



Et lorsqu'il n'y a aucune place disponible dans le parking le visuel suivant :



## V. La boucle de courant

La boucle de courant 4-20 mA est un moyen de transmission permettant de transmettre un signal analogique sur une grande distance sans perdre d'information sur le signal. Cette boucle répond au besoin de transmettre un signal analogique et ce depuis le premier capteur. Au début, les ingénieurs ont eu de grandes difficultés à trouver un moyen de transmission fiable, car la transmission filaire induisait une perte des données. Deux solutions furent proposées : la première par transmission d'impulsion PDM (Pulse Duration Modulation) et l'autre par variations proportionnelles d'une fréquence selon la valeur analogique. Mais ces deux solutions n'étaient pas optimales, du fait de leurs coûts élevés. Lorsque la boucle 4-20 mA arriva sur le marché elle est vite devenue le standard car elle n'était pas affectée par la variation de longueur des fils et par la variation des résistances.

Pour réaliser une boucle 4-20 mA il faut au moins quatre éléments qui sont l'émetteur, l'alimentation de la boucle, les fils de la boucle et le récepteur.

L'émetteur est composé d'un capteur qui va mesurer les grandeurs physiques telles que la pression, la température etc ... et d'un émetteur de courant 4-20 mA. L'émetteur convertit la grandeur physique en courant compris dans l'intervalle 4-20 mA ou 4 mA. Elle sera associée à la valeur minimale que l'on doit mesurer et 20 mA sera associé à la plus grande valeur. (Par exemple si on doit mesurer une température comprise entre -40 et 60 degrés 4 mA sera associé à -40 degrés et 20 mA sera associé à 60 degrés) Si on lit 0 mA c'est que la boucle ne fonctionne pas ou qu'il y a une erreur.

L'émetteur doit être alimenté pour fonctionner, ceci est réalisé grâce aux deux fils de la boucle, deux fils relient tous les composants entre eux. Il y a 4 conditions pour le choix de ces fils :

- Très faible résistance
- Bonne protection contre la foudre
- Ne pas subir d'impulsion de tension
- Avoir une seule mise à la masse.

Il y a toujours au moins un récepteur dans la boucle. Il peut s'agir d'un afficheur digital, une table d'enregistrement, un déclencheur de vannes ... mais ils ont tous une

chose en commun: une résistance. On peut mettre autant de récepteur que l'on veut à condition qu'il y ait assez de tension pour alimenter la boucle.

## VI. Conclusion

La réalisation de ce projet m'a beaucoup apporté tant sur le plan technique que sur le plan humain, car un projet de groupe nécessite des qualités humaines que je n'avais pas particulièrement développé. J'ai appris à écouter les autres et tenir compte des remarques qui me sont faites bonnes ou mauvaises. J'ai également progressé sur le plan technique notamment dans le domaine du développement web.

Enfin j'ai bien aimé travailler sur ce projet car le travail de groupe permet une mise en commun des connaissances et des compétences de chacun. Cela permet de ne pas rester bloqué sur un problème longtemps car les autres membres du groupe peuvent venir en aide à celui qui a un problème.