DOCUMENTATION PARKING

Table des matières

[Introduction : 1](#_Toc21353376)

[1- Création et mise en place des données : 2](#_Toc21353377)

[1.1 Modélisation logique de la base de données (MLD) 2](#_Toc21353378)

[1.2 : Modèle de conception des données (E/A) 2](#_Toc21353379)

[2- Securité des BDD 3](#_Toc21353380)

[3- Mise en place de l’espace administrateur 3](#_Toc21353381)

[4- Mise en place de l’espace utilisateur 4](#_Toc21353382)

[5- Mise en place de la page WEB 4](#_Toc21353383)

# Introduction :

Afin d’éviter le stationnement sauvage dans le labyrinthe qu’est le parking, il a été décidé d’attribuer à chaque membre qui le demandait une place de parking numérotée.

Voici les objectifs demandés afin d’éviter le stationnement sauvage dans le labyrinthe qu’est le parking.

* Le front-office doit être sécurisé et n’accepter que les demandes du personnel des ligues. Les inscriptions au service de réservation de place doivent être validées (ou créées) par un administrateur.
* L’administrateur, seul utilisateur du back-office, doit pouvoir éditer la liste des places et gérer les inscriptions des utilisateurs.
* Lorsqu’un utilisateur en fait la demande, une place libre lui est attribuée aléatoirement et immédiatement par l’application, la réservation expire automatiquement au bout d’une durée par défaut déterminée par l’administrateur.
* Si une demande ne peut pas être satisfaite, l’utilisateur est placé en liste d’attente.
* Un utilisateur ou l’administrateur peuvent fermer une réservation avant la date d’expiration prévue. Une fois celle-ci expirée, l’utilisateur doit refaire une demande s’il souhaite obtenir une place.

## Création et mise en place des données :

### Modélisation logique de la base de données (MLD)

### #------------------------------------------------------------

### # Script MySQL.

### #------------------------------------------------------------

### #------------------------------------------------------------

### # Table: User

### #------------------------------------------------------------

### CREATE TABLE User(

### id\_user Int Auto\_increment NOT NULL ,

### nom\_user Varchar (50) NOT NULL ,

### Prenom\_user Varchar (50) NOT NULL ,

### admin\_user Varchar (150) NOT NULL ,

### mdp\_user Varchar (150) NOT NULL ,

### place Int NOT NULL

### ,CONSTRAINT User\_PK PRIMARY KEY (id\_user)

### )ENGINE=InnoDB;

### #------------------------------------------------------------

### # Table: Place

### #------------------------------------------------------------

### CREATE TABLE Place(

### id\_place Int Auto\_increment NOT NULL ,

### datereservation Date NOT NULL

### ,CONSTRAINT Place\_PK PRIMARY KEY (id\_place)

### )ENGINE=InnoDB;

### #------------------------------------------------------------

### # Table: Date

### #------------------------------------------------------------

### CREATE TABLE Date(

### date Time NOT NULL

### ,CONSTRAINT Date\_PK PRIMARY KEY (date)

### )ENGINE=InnoDB;

### #------------------------------------------------------------

### # Table: Reserver

### #------------------------------------------------------------

### CREATE TABLE Reserver(

### id\_user Int NOT NULL ,

### id\_place Int NOT NULL ,

### date Time NOT NULL

### ,CONSTRAINT Reserver\_PK PRIMARY KEY (id\_user,id\_place,date)

### ,CONSTRAINT Reserver\_User\_FK FOREIGN KEY (id\_user) REFERENCES User(id\_user)

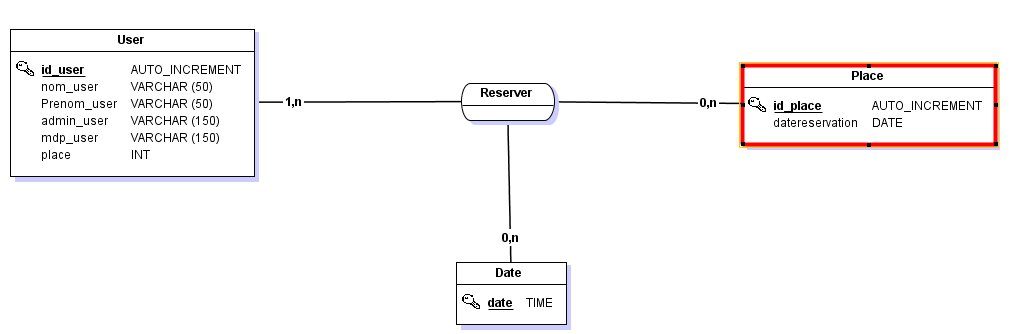
### ,CONSTRAINT Reserver\_Place0\_FK FOREIGN KEY (id\_place) REFERENCES Place(id\_place)

### ,CONSTRAINT Reserver\_Date1\_FK FOREIGN KEY (date) REFERENCES Date(date)

### )ENGINE=InnoDB;

### : Modèle de conception des données (E/A)

Afin de mieux visualiser cela voici un modèle de conception des données (MCD) de l’architecture de notre base de données :

Num file d’attente dans utilisateur

## Securité des BDD

Afin de securisé nos bases de données(BDD) afin que les utilisateurs puisse se connecter et reserver en toute securité nous allons proceder en plusieurs etapes qui sont :

* Protection des accès par mot de passe.
  + Hachage des mots de passe
  + Fonction "mot de passe perdu ?"
* Contrôles de saisie des données côté serveur
* Contrôles de saisie côté client
* Protection contre les attaques par injection

Attaque par injection :

**Une requête SQL est une demande d'action à effectuer dans une base de données, le plus souvent dans une page web qui demande un nom d'utilisateur ou un mot de passe. Mais, la plupart des sites web ne surveillant pas les entrées autres que les noms d'utilisateur et les mots de passe, un pirate informatique peut utiliser les zones d'entrée pour envoyer ses propres requêtes, c'est-à-dire injecter du SQL dans la base de données. De cette manière, les pirates informatiques peuvent créer, lire, mettre à jour, altérer ou supprimer les données stockées dans la base de données principale, généralement pour accéder à des informations sensibles comme les numéros de sécurité sociale et les données de cartes de crédit ainsi que d'autres informations financières.**

## Mise en place de l’espace administrateur

* Protection de l’accès par mot de passe
* Édition de la liste des utilisateurs, réinitialisation des mots de passe
* Édition de la liste des places
* Consultation de la liste d’attente.
* Consultation de l’historique d’attribution des places.
* Attribution manuelle des places.
* Édition de la file d’attente (modification de la position des personnes en attente).

## Mise en place de l’espace utilisateur

* Vérification de l’identité par saisie d’un mot de passe
* Possibilité de visualiser le numéro de place attribuée, ainsi que l’historique des places précédemment attribuées
* Possibilité de faire une demande de réservation
* Possibilité de connaître son rang sur la file d’attente
* Modification du mot de passe

## Mise en place de la page WEB

Le CSS (Cascading Style Sheets) correspond à un langage informatique permettant de mettre en forme des pages web (HTML ou XML).

HTML (HyperText Markup Language) est un langage informatique utilisé sur l'internet. Ce langage est utilisé pour créer des pages web.

PHP (PHP : Hypertext Preprocessor) est un langage est principalement utilisé pour produire un site web dynamique

Nous allons créer notre page web à l’aide du CSS et du HTLM ainsi que du PHP afin de rendre notre page dynamique et plus particulièrement pour sécuriser les mots de passe. Le HTML va nous permettre de créer notre page web et le CSS de le mettre en forme, c’est-à-dire lui donner le style désirer. Le Responsive Web Design (RWD) que nous allons mettre en place grâce au HTML et CSS sert à ajuster automatiquement l’affichage d’une page web à la taille d’écran du terminal utilisé (Portable, ordinateur.).

https ://page.user/edit/..

ESPACE UTILISATEUR

* Possibilité de visualiser le numéro de place attribuée, ainsi que l’historique des places précédemment attribuées.
* Possibilité de faire une demande de réservation.
* Possibilité de connaître son rang sur la file d’attente.
* Modification du mot de passe.

Voici ci-dessous une représentation de l’architecture de l’application :

https ://page.user/..

ESPACE UTILISATEUR

* Vérification de l’identité par saisie d’un mot de passe.
* Possibilité de faire une demande de réservation.

https ://page.accueil/..

Login

password

Login

password

Espace Utilisateurs

Espaces administrateurs

https ://page.admin/..

https ://page.admin/edit/..

ESPACE ADMINISTATEUR

* Consultation de l’historique d’attribution des places.
* Édition de la liste des utilisateurs, réinitialisation des mots de passe.
* Édition de la liste des places
* Consultation de la liste d’attente.
* Édition de la file d’attente (modification de la position des personnes en attente).

https ://page.admin/..

ESPACE ADMINISTATEUR

* Protection de l’accès par mot de passe.
* Attribution manuelle des places.

Liste des URL :

https ://page.accueil/..

https ://page.admin/..

https ://page.user/..

https ://page.user/edit/..

https ://page.accueil/edit/..