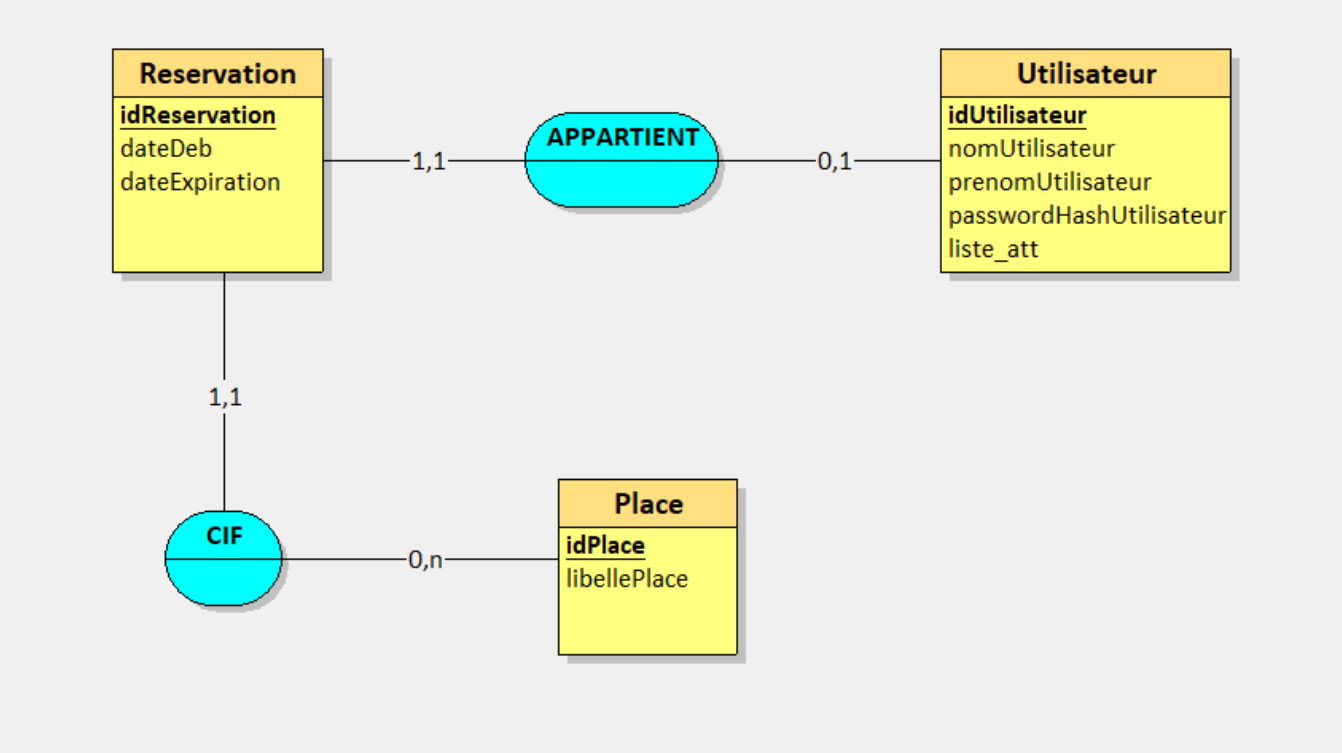
Documentation Parking

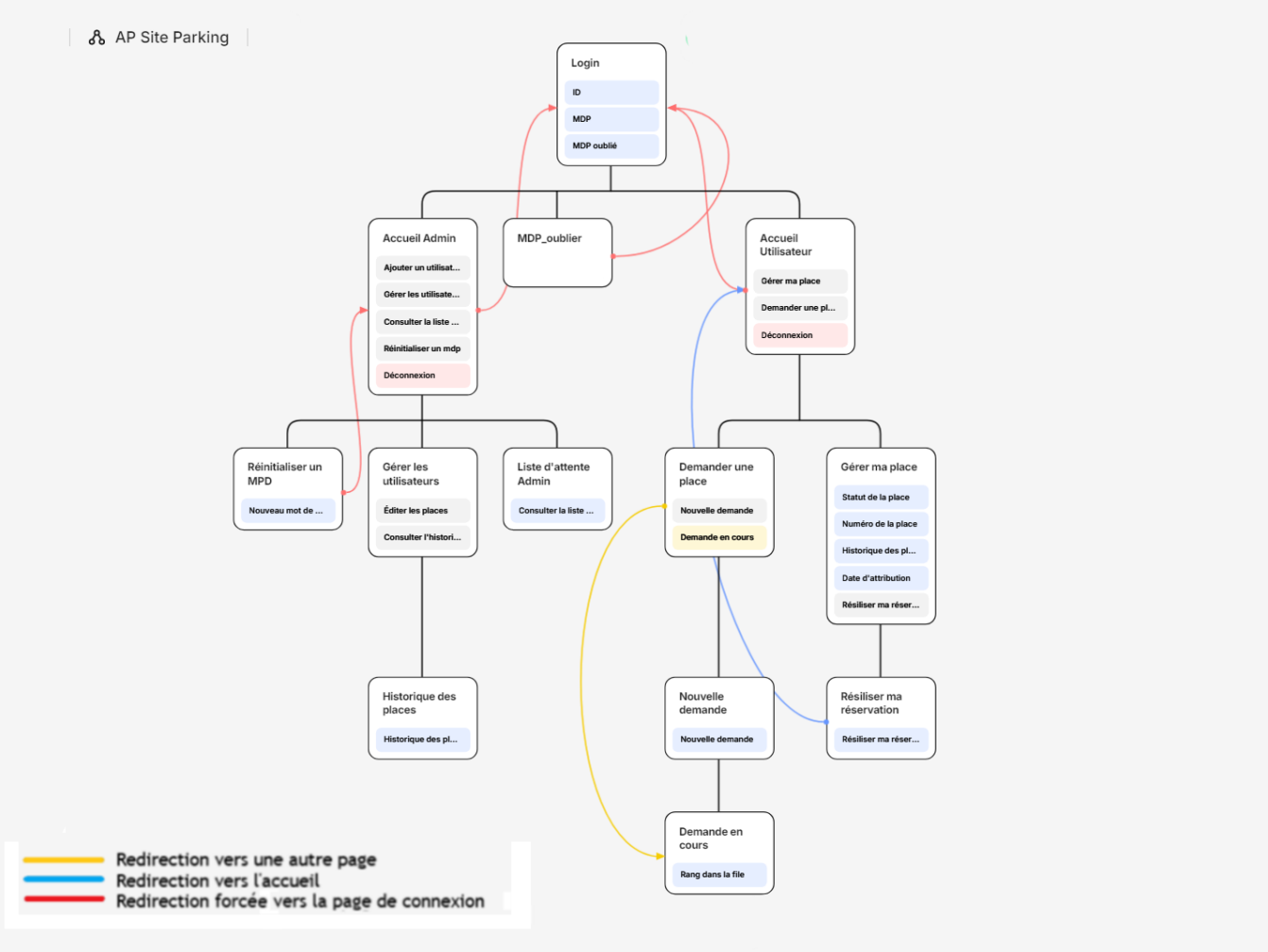
I / Création du MCD et de la maquette

MCD :

- Une réservation concerne une et une seul place
- Une réservation appartient a un et un seul utilisateur
- Un utilisateur peut avoir aucune ou une réservation
- Une place est concerné par aucune ou plusieurs réservation



Maquette :


AP Site Parking
Login
ID
MDP
MDP oublié
Accueil Admin
Ajouter un utilisat...
Gérer les utilisate...
Consulter la liste ...
Réinitialiser un mdp
Déconnexion
MDP_oublier
Accueil Utilisateur
Gérer ma place
Demander une pl...
Déconnexion
Réinitialiser un MPD
Nouveau mot de ...
Gérer les utilisateurs
Éditer les places
Consulter l'histori...
Liste d'attente Admin
Consulter la liste ...
Demander une place
Nouvelle demande
Demande en cours
Gérer ma place
Statut de la place
Numéro de la place
Historique des pl...
Date d'attribution
Résiliser ma réser...
Historique des places
Historique des pl...
Nouvelle demande
Nouvelle demande
Résiliser ma réservation
Résiliser ma réser...
Demande en cours
Rang dans la file
Redirection vers une autre page
Redirection vers l'accueil
Redirection forcée vers la page de connexion

II / Création de la base de données

1) Création de la table users
2)

```php
public function up(): void
{
    Schema::create('users', function (Blueprint $table) {
        $table->id();
        $table->string('nom');
        $table->string('prenom');
        $table->string('email')->unique();
        $table->timestamp('email_verified_at')->nullable();
        $table->string('password');
        $table->boolean('role');
        $table->integer('listeatt')->nullable();
        $table->rememberToken();
        $table->timestamps();
    });
```

Rôle :

0 = Employé
1 = Administrateur

```php
public function isAdmin(){
    return $this->role == 1;
}
```

3) Création de la table places

```php
public function up(): void
{
    Schema::create('places', function (Blueprint $table) {
        $table->id();
        $table->string('libellePlace');
        $table->string('status');
        $table->timestamps();
    });
}
```

4) Création de la table réservations

```php
public function up(): void
{
    Schema::create('reservations', function (Blueprint $table) {
        $table->id();
        $table->integer('user_id');
        $table->integer('place_id')->nullable();
        $table->date('dateDeb')->nullable();
        $table->date('dateExpiration')->nullable();
        $table->date('dateDemande');
        $table->integer('status');
        $table->timestamps();
    });
}
```

5) Création d'un seeder

```
    User::factory()->create([
        'nom' => 'test',
        'prenom' => 'test',
        'email' => 'a@a',
        'password' => 'a',
        'role'=>1,
        'listeatt'=>1,
    ]);

    User::factory()->create([
        'nom' => 'test',
        'prenom' => 'test',
        'email' => 'b@b',
        'password' => 'a',
        'role'=> 0
    ]);

    place::create([
        'libellePlace' => "placeTest",
        'status' => 'libre',
    ]);


    reservation::create([
        'user_id' => 1,
        'place_id' => 1,
        'status' => 0,
        'dateDeb' => Carbon::now()->format('Y-m-d'),
        'dateExpiration' => Carbon::now()->addWeeks(3)->format('Y-m-d'),
        'dateDemande' => Carbon::now()->format('Y-m-d'),
    ]);
}
```

III / Controllers/Middlewares/Routes

Pour le système d'authentification nous avons utilisé une bibliothèque nommé Laravel Breeze qui se chargera de cela

Reservation Controller :

Routes	Methode HTTP	Description
reservations/	POST	Permet d'enregistrer une reservations en base de données (reservations.store)
reservations /{i}/edit	GET	Permet d'afficher le formulaire de modifications d'une reservations (reservations.edit)
reservations /{i}	PUT/PATCH	Permet de modifier le reservations i (reservations.update)
reservations/{i}	DELETE	Permet de supprimer un reservations i (reservations.destroy)

Administrateur Controller :

Routes	Methode HTTP	Description
admin/	GET	Affiche tous les utilisateurs (admin.index)
admin/create	GET	Permet d'afficher le formulaire pour créer des utilisateurs (admin.create)
admin	POST	Permet d'enregistrer un ou plusieurs utilisateurs en base de données (admin.store)
admin/{i}/edit	GET	Permet d'afficher le formulaire de modifications d'un utilisateur (admin.edit)
admin/{i}	PUT/PATCH	Permet de modifier le utilisateur i (admin.update)
admin/{i}	DELETE	Permet de supprimer un utilisateur i (admin.destroy)

La route **admin/create** permet de créer plusieurs utilisateurs en spécifiant uniquement leurs adresses email, séparées par une virgule ou un espace.

place Controller :

Routes	Methode HTTP	Description
place/	GET	Affiche tous les place (place.index)
place/create	GET	Permet d'afficher le formulaire de créatio d'une place(place.create)
place/	POST	Permet d'enregistrer une place en base de données
Place/(i}/edit	GET	Permet d'afficher le formulaire de modification d'une place
Place/{i}	PUT/PATCH	Permet d'enregistrer les modifications en base de données
Place/{i}	DELETE	Permet de supprimer la place de la DB en résiliant la réservations

IV / Les events , leurs listeners et notifications

Event	Listener	Notifications	Description
creerUserEvent	creerUserListener	creerUserNotification	Sert à envoyer un mail à tous les utilisateurs dont le compte vient d'être créer pour réinitialisé leur mdp.
reserverEvent	reserverEvent		Sert a gérer la liste d'attente.

Toutes ces notifications implémente ShouldQueue pour effectuer ces taches en arrière-plan

V / Les policies et vues.

reservation Policy :

Policy	Droit
Resilier en fonction d'une reservation	L'utilisateur est propriétaire de la reservation ou est Administrateur
choisir	Seul un administrateur et que le nombre de place dispo different de 0
Create	Si l'utilisateur n'a aucune reservation en liste d'attente ou actif
Update	Seul un administrateur
Attribuer place	Seul un administrateur

place Policy :

Policy	Droit
viewAny	Seul un administrateur
Create	Seul un administrateur
Update	Seul un administrateur
Delete	Seul un administrateur

User Policy :

Policy	Droit
View	Seul un administrateur
Create	Seul un administrateur
Update	Seul un administrateur
Delete	Seul un administrateur

Admin view :

Vue	Description
main	Afficher tous les utilisateurs et leurs rôles.
edit	Afficher le formulaire de modification d’un utilisateur
createUsers	Afficher le formulaire de création de plusieurs utilisateurs.

reservations view :

Vue	Description
main	Afficher toutes les reservations en attente.
edit	Afficher le formulaire de modification d’une reservation.
prise	Afficher toutes les reservations occupé.

Employé view :

Vue	Description
main	Afficher l’espace de l’employé avec ces ancienne réservation .

Place view :

Vue	Description
main	Afficher toutes les places.
edit	Afficher le formulaire de modification d’une place.
create	Afficher le formulaire de création d’une place.

Conclusion :

En conclusion, ce projet m’a permis d’approfondir mes connaissances en Laravel en mettant en pratique des concepts clés tels que la gestion des routes, l’utilisation des Eloquent Models, la création de migrations et l’implémentation de fonctionnalités complexes. Cette expérience m’a permis de renforcer ma compréhension du framework et d’acquérir une meilleure maîtrise des bonnes pratiques de développement, tout en améliorant ma capacité à résoudre des problèmes et à structurer efficacement le code.