

DOCUMENTATION TECHNIQUE

Sujet Groupe Hôtelier - Session Juin 2022

AITOR GARCÍA BELLVER

1. Spécifications techniques

Pour le front-end:

- HTML5
- CSS3
- SASS
- Javascript / JQuery
- Webpack encore
- Twig
- Bootstrap 5
- Node.js

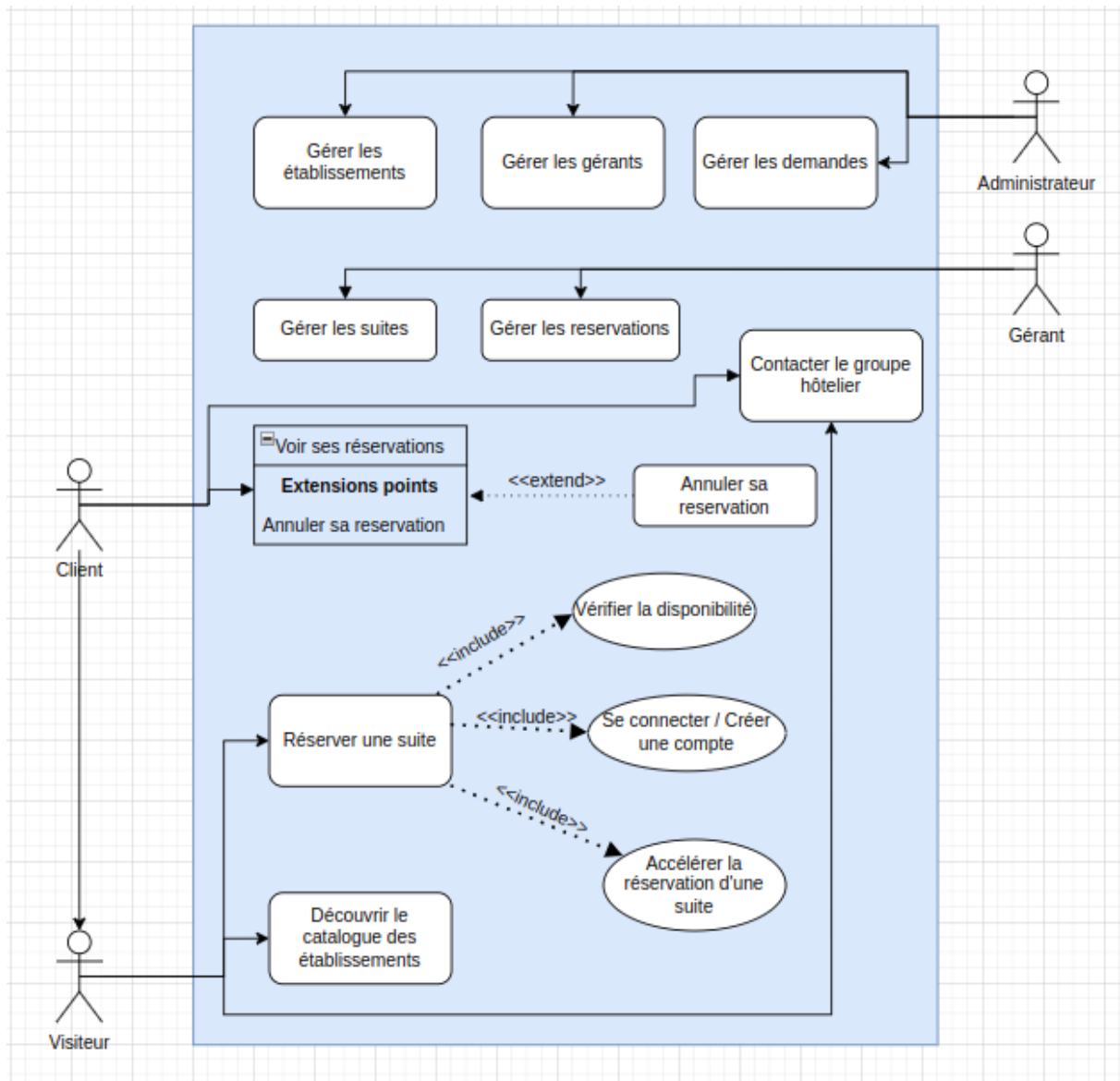
Pour le back-end:

- Symfony 6.0.
- Easyadmin 4 symfony/bundle
- PHP 8.1
- Vich uploader symfony/bundle

Pour le déploiement:

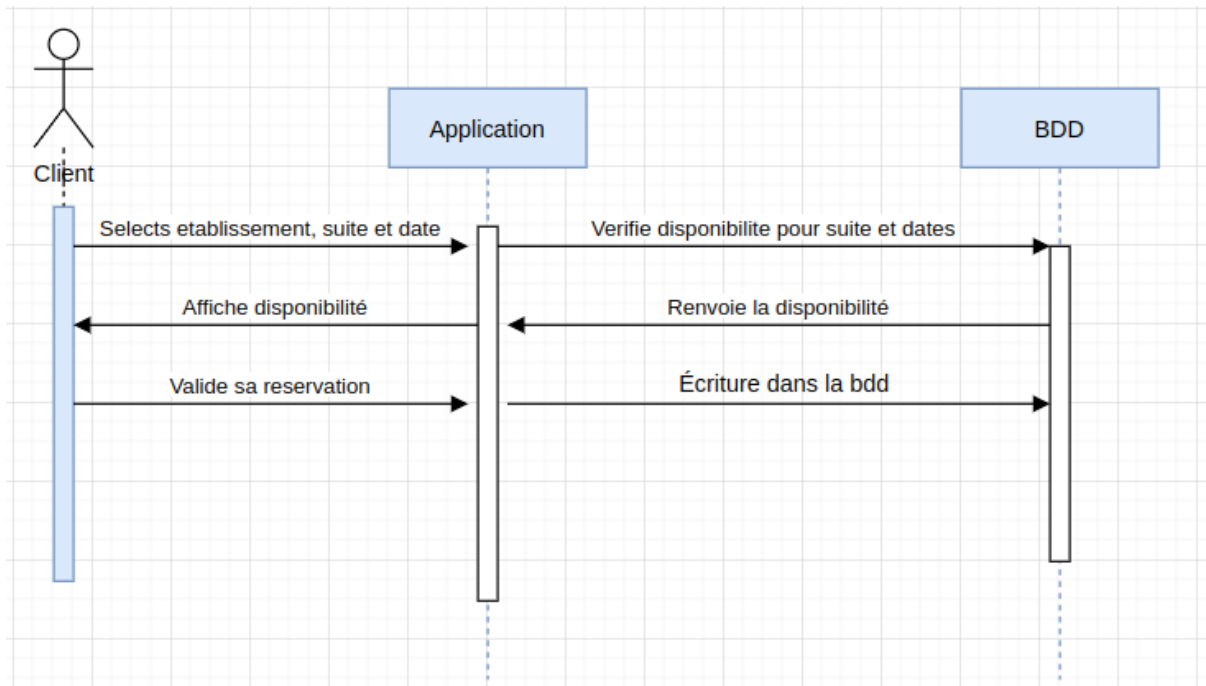
- Heroku
- Heroku POSTGRES (as database)
- Apache / Debian
- Extension PHP: intl, pgsql
- PHP VERSION: 8.1

2. Diagramme cas d'utilisation

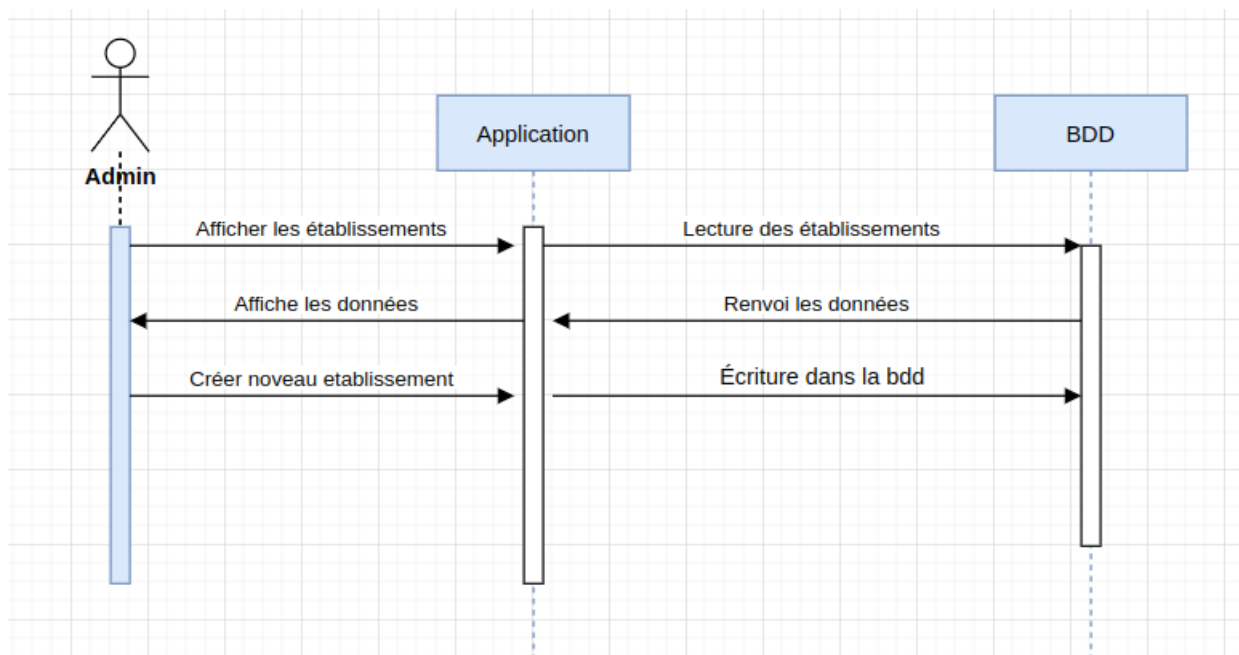


3. Diagrammes de séquence

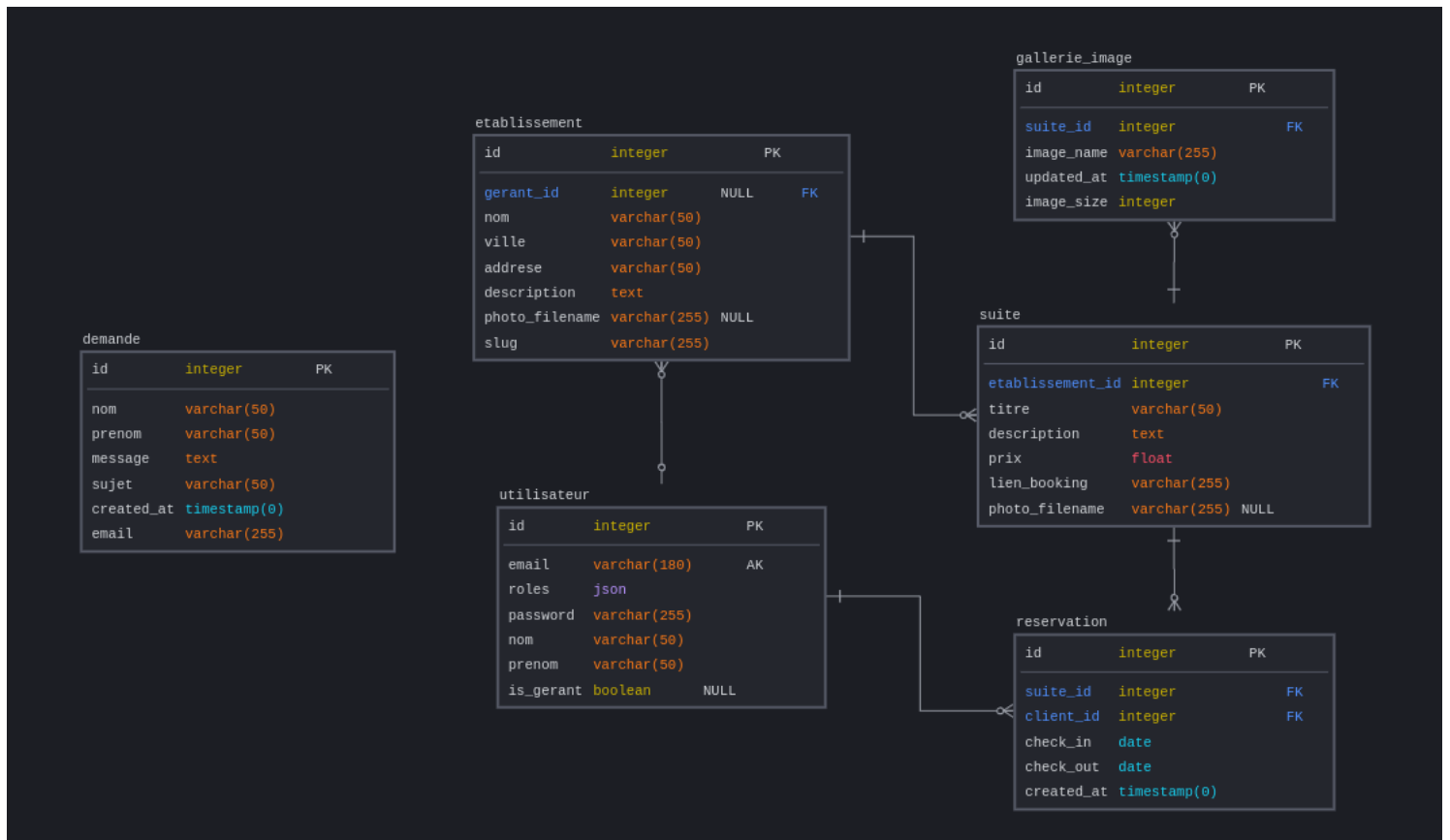
3.1. US4 - Réserver une suite



3.2. US1 - Gérer les établissements



4. Modèle physique des données



5. Fonctionnement de la vérification de disponibilité d'une suite

La vérification de disponibilité d'une suite se déroule en deux étapes.

Tout d'abord, nous utilisons jquery pour collecter les données saisies par l'utilisateur dans le formulaire de réservation. Comme vous pouvez le voir à la **ligne 101**, les données seront modifiées dynamiquement chaque fois que l'utilisateur change la suite, la date d'arrivée ou la date de départ. En même temps, à la **ligne 108**, vous pouvez voir que la requête ajax ne sera lancée que lorsque les trois champs (suite, checkin, checkout) auront été remplis.

```
101 $(document).on('change', '#reservation_suite, #reservation_checkIn, #reservation_checkOut', function(){
102     let suite = $('#reservation_suite').val()
103     let checkIn= $('#reservation_checkIn').val()
104     let checkOut= $('#reservation_checkOut').val()
105     let hotelText = $('#reservation_etablissement option:selected').text()
106     let suiteText = $('#reservation_suite option:selected').text()
107
108     if (suite && checkIn && checkOut) {
109
```

assets/app.js

Une fois les données remplies, on lance une requête ajax qui prend comme data la suite, le checkin et le checkout

En fonction de la réponse reçue à la demande, nous allons afficher:

- un message d'alerte à l'utilisateur (dates invalides, **lignes 148-152**)
- un message en disant que la suite n'est pas disponible (**lignes 141-147**)
- On va montrer à l'utilisateur que la suite est disponible et débloquent le bouton de réservation pour qu'il puisse la valider(**ligne 154 +**).

```
131     let data={}
132     data['suite'] = suite
133     data['checkIn'] = checkIn
134     data['checkOut'] = checkOut
135
136     $.ajax({
137         type: 'POST',
138         url: '/reservation/check_availability',
139         data: data,
140         success: function(data){
141             if(data.status === 'error'){
142                 $('#submit_reservation').attr('disabled', true)
143                 $('#reservation_checkIn').addClass('invalid')
144                 $('#reservation_checkOut').addClass('invalid')
145                 $('#availability_info')
146                     .html('<strong class="red">Les dates choisies ne sont pas disponibles')
147             } else if (data.status === 'invalid_dates'){
148                 $('#reservation_checkIn').addClass('invalid')
149                 $('#reservation_checkOut').addClass('invalid')
150                 $('#availability_info')
151                     .html('<strong>Les dates choisies ne sont pas disponibles')
152             } else {
153                 $('#submit_reservation').attr('disabled', false)
154                 $('#availability_info')
155                     .html('<strong>La suite est disponible')
156             }
157         }
158     })
159 }
```

La requête ajax est reçu par le ReservationController, qui exécute les opérations suivantes:

```
43      #[Route('/check_availability', name: '_check_availability', methods: ['GET', 'POST'])]
44
45      public function checkAvailability(ReservationRepository $reservationRepository, Request $request)
46      {
47          $suite = $request->request->get( key: 'suite');
48          $checkIn = $request->request->get( key: 'checkIn');
49          $checkOut = $request->request->get( key: 'checkOut');
50          $checkInDate = strtotime($checkIn);
51          $checkOutDate = strtotime($checkOut);
52          $yesterday = time() - (86400);
53
54          if (!empty($suite) && !empty($checkIn) && !empty($checkOut)) {
55
56              if ($checkInDate >= $checkOutDate || $checkInDate <= $yesterday) {
57
58                  return $this->json([
59                      'code' => 200,
60                      'status' => 'invalid_date',
61                      'message' => 'Vous pouvez pas voyager dans le temps'
62                  ], status: 200);
63
64              } else {
65                  $reservations = $reservationRepository->isAvailable($suite, $checkIn, $checkOut);
66
67                  if (count($reservations) === 0) {
68
69                      return $this->json([
70                          'code' => 200,
71                          'status' => 'success',
72                          'message' => 'Date disponible'
```

- Reçoit les paramètres de la requête (lignes 47-49) et vérifie si tous sont remplis avant de continuer (ligne 54).
- On vérifie si la date de check in est inférieure à celle du checkout et que la date de check in soit, au moins, le jours en cours. Si c'est pas le cas, en retourne un status "invalid_date"
- Si la requête est valide, on va utiliser la fonction isAvailable (ligne 65) du ReservationRepository pour retrouver les reservation pour ces dates dans cette suite. Après on va compter le total de reservations, et si le total est zero, on retourne un status "succes" indiquant que la suite es disponible.
- Sinon on va renvoyer un status "error" en indiquant que la suite n'est pas disponible (image suivant)

```
    } else {
        return $this->json([
            'code' => 200,
            'status' => 'error',
            'message' => 'Date indisponible'
        ], status: 200);
    }
```

Annexe: Function `isAvailable()` dans le `ReservationRepository`.

```
48 public function isAvailable($suite, $checkIn, $checkOut)
49 {
50     $query = $this->createQueryBuilder( alias: 'r')
51         ->andWhere('r.suite = :suite AND r.checkIn BETWEEN :checkIn AND :checkOut')
52         ->orWhere('r.suite = :suite AND r.checkOut BETWEEN :checkIn AND :checkOut')
53         ->setParameter( key: 'suite', $suite )
54         ->setParameter( key: 'checkIn', $checkIn )
55         ->setParameter( key: 'checkOut', $checkOut )
56         ;
57     return $query->getQuery()->getResult();
58 }
```

On lance un query pour recueillir toutes les réservations ou la suiteId est égal à la suiteId on paramètre et ou il y ait des checkIns ou checkOuts pour les dates passées en parametre.