

Documentation technique



A. Spécifications techniques

Serveur :

Heroku

Version PHP 8.2

Extension PHP : PDO

ClearDB

Pour le front :

HTML 5

CSS 3

Bootstrap 5

JavaScript

Pour le back :

PHP 8.2 sous PDO

Symfony 6.2

MySQL

Compétences :

- Maquetter une application
- Réaliser une interface utilisateur web statique et adaptable
- Développer une interface utilisateur web dynamique
- Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce- Créer une base de données
- Développer les composants d'accès aux données
- Développer la partie back-end d'une application web ou web mobile
- Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce

Choix des technologies front-end:

Pour la partie front-end, j'ai décidé d'utiliser HTML5, CSS3, Bootstrap et JavaScript.

Le design de mon application est basé sur des cartes (pour l'affichage des services et des annonces), Bootstrap permet de gérer simplement le positionnement et le côté responsive.

Outre le positionnement et le côté responsive, Bootstrap propose un large éventail de classes très utiles telle que la barre de navigation.

CSS3 me permet d'apporter un côté personnalisé et d'appliquer des media-queries plus spécifiques.

Enfin, les visiteurs doivent pouvoir filtrer sur les annonces sans rechargement de page, d'où le choix d'utiliser JavaScript avec une requête Ajax.

JavaScript permet de manipuler les éléments du DOM : ajuster les valeurs lorsque l'utilisateur applique un filtre numérique ou sélectionner/désélectionner toutes les checkbox en cliquant sur « Tout sélectionner ».

Enfin la requête Ajax (utilisation de fetch), me permet d'effectuer une requête filtrée vers la BDD et d'afficher les résultats sans rechargement de la page.

Choix des technologies back-end:

Pour la partie back-end, j'ai décidé d'utiliser PHP 8.2 sous PDO, Symfony 6.2 et MySQL.

J'utilise Symfony version 6.2, même si la version 6.3 est disponible depuis mai 2023, car la version 6.2 a eu le temps d'être testée par d'autres développeurs, les problèmes remontés et il est plus facile de trouver des solutions sur les forums.

Symfony dispose de nombreux bundles très utiles.

Pour la gestion du back-end, j'ai installé le bundle EasyAdmin qui permet de gérer tous les éléments (annonces, utilisateurs, ...) au même endroit. D'autres bundles ont été installés sur le projet, tels que VichUploader ou Fixtures.

Niveau sécurité, le bundle Security permet de se protéger contre les attaques CSRF. J'ai également intégré Google reCAPTCHA au formulaire de contact, des contraintes sont appliquées pour la création d'un mot de passe (12 caractères dont 1 majuscule, 1 minuscule, 1 caractère spécial, 1 chiffre, selon les recommandations du CNIL) et les routes sont sécurisées selon le rôle de l'utilisateur.

De plus, le Profiler de Symfony permet d'identifier plus facilement d'où vient l'erreur, ce qui permet un gain de temps.

Le choix de MySQL s'explique tout d'abord par sa grande communauté. En cas de problème, il est plus simple de trouver une solution sur les forums dans la documentation.

MySQL est simple d'utilisation et plus rapide que la concurrence.

Choix des technologies pour serveur:

Ayant déjà utilisé Heroku, je suis satisfait de leurs services. En effet, leur service client est très réactif avec des réponses à 1 jour ouvré à chacune de mes demandes.

De plus, le déploiement en ligne est très simple via Github.

Sur Heroku, il existe plusieurs choix pour la base de données, la plus populaire étant ClearDB. Son utilisation est simple, les ressources sur internet sont nombreuses et ses performances sont reconnues au sein de la communauté.

Configuration de l'environnement de travail :

1. Installation de Symfony

Via la commande : `symfony new garage-parrot --version=6.2 -webapp`

2. Utilisation de XAMPP

Téléchargement et installation

Ajout du chemin des exécutables PHP et MySQL dans les variables d'environnement

Lancement du serveur Apache et MySQL

3. Configuration du fichier .env pour accéder à la base de données en local

- Lancer la commande `php bin/console database :create`

4. Utilisation de composer

- Téléchargement sur <https://getcomposer.org/download/>

- Installation

5. Utilisation de Git

- Téléchargement de Git

- Installation

- Création d'un repository sur Git

- Initialisation du projet Git sur VSCode à la racine du projet et liaison avec le dépôt distant

6. Utilisation de Bootstrap sans CDN

- Téléchargement et import de Bootstrap dans le projet

7. Utilisation Heroku

- Téléchargement et installation de Heroku CLI

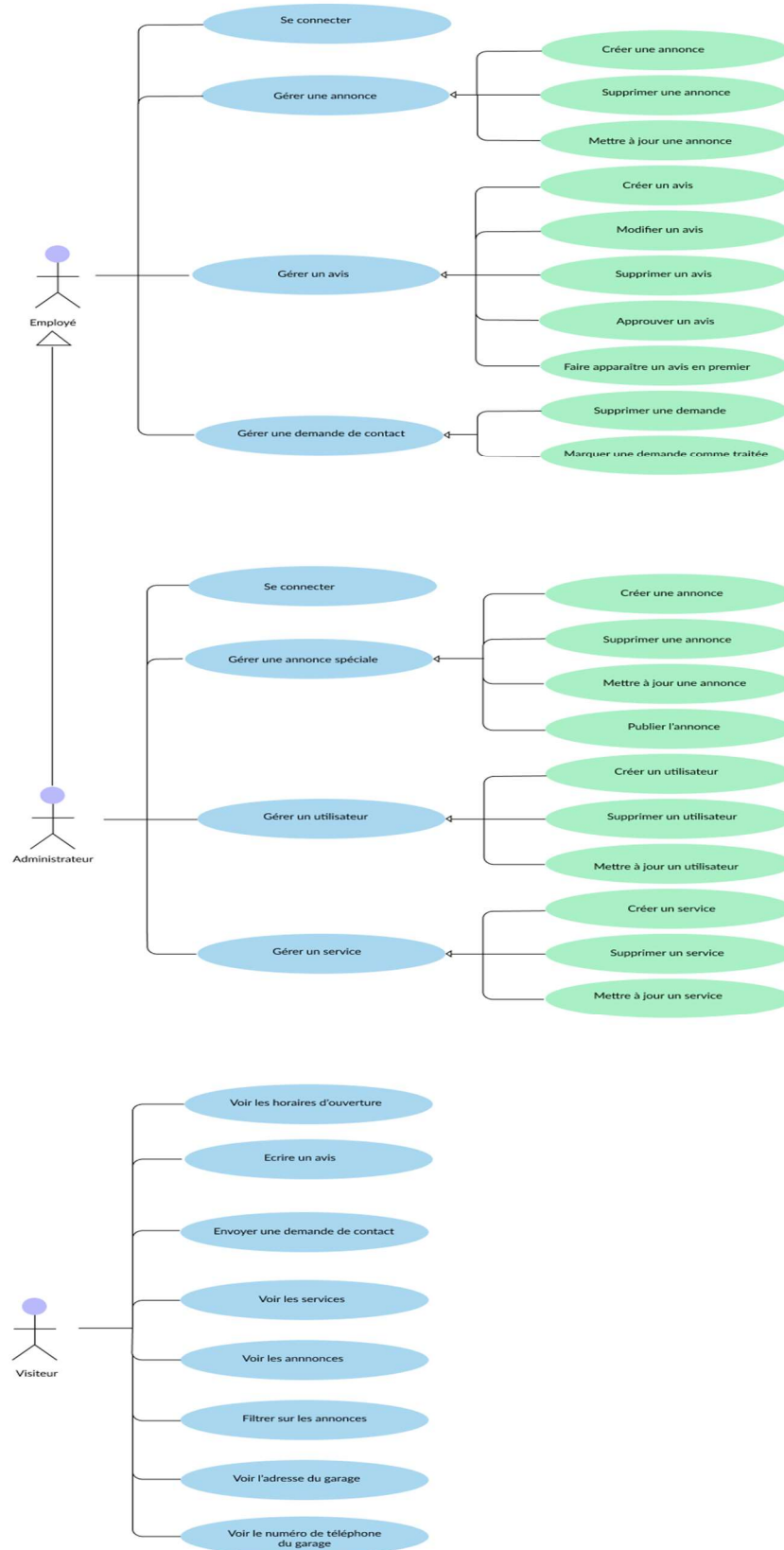
- Sur Heroku, liaison avec le projet GitHub

-Initialisation d'Heroku sur le projet

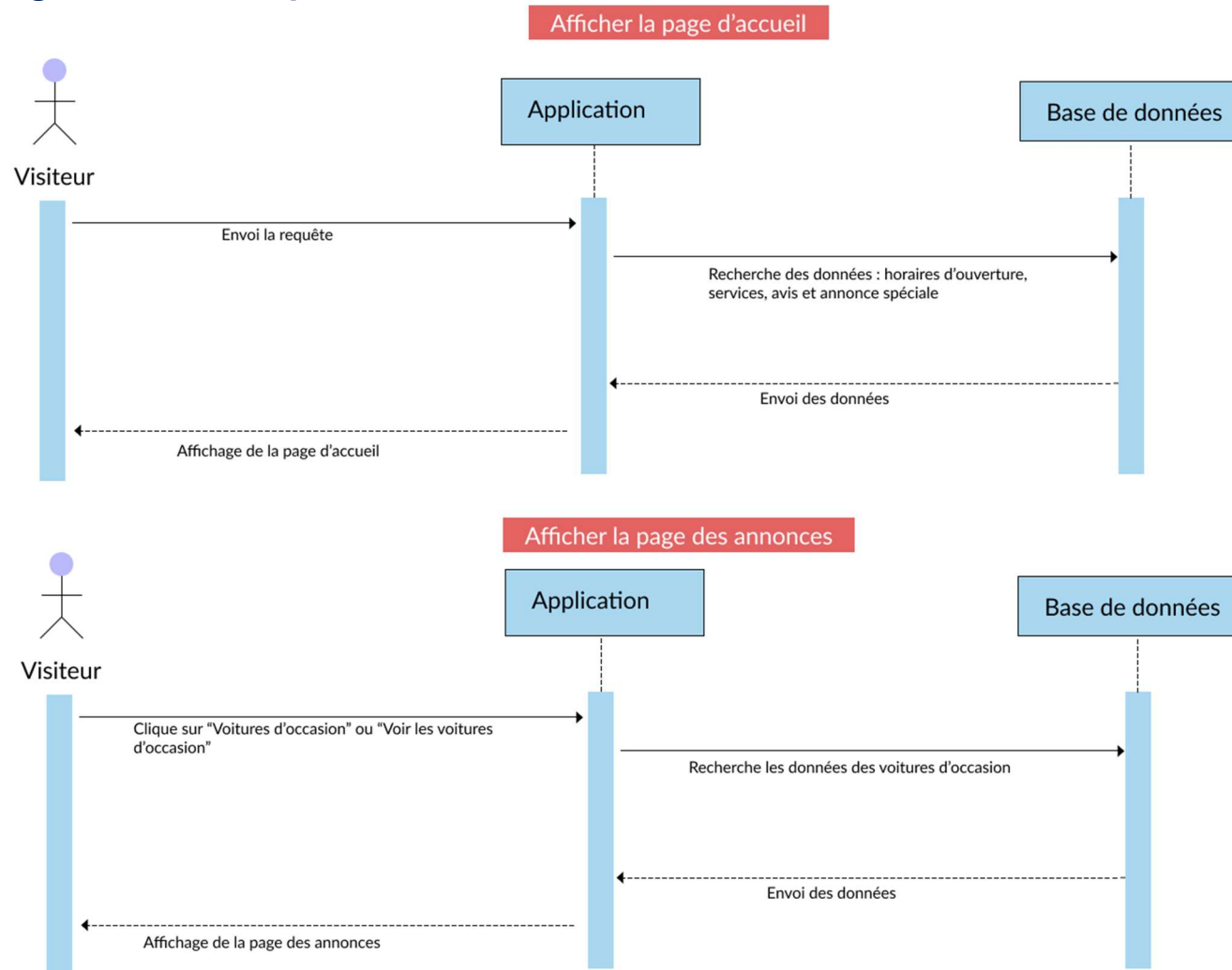
8. Utilisation de ClearDB

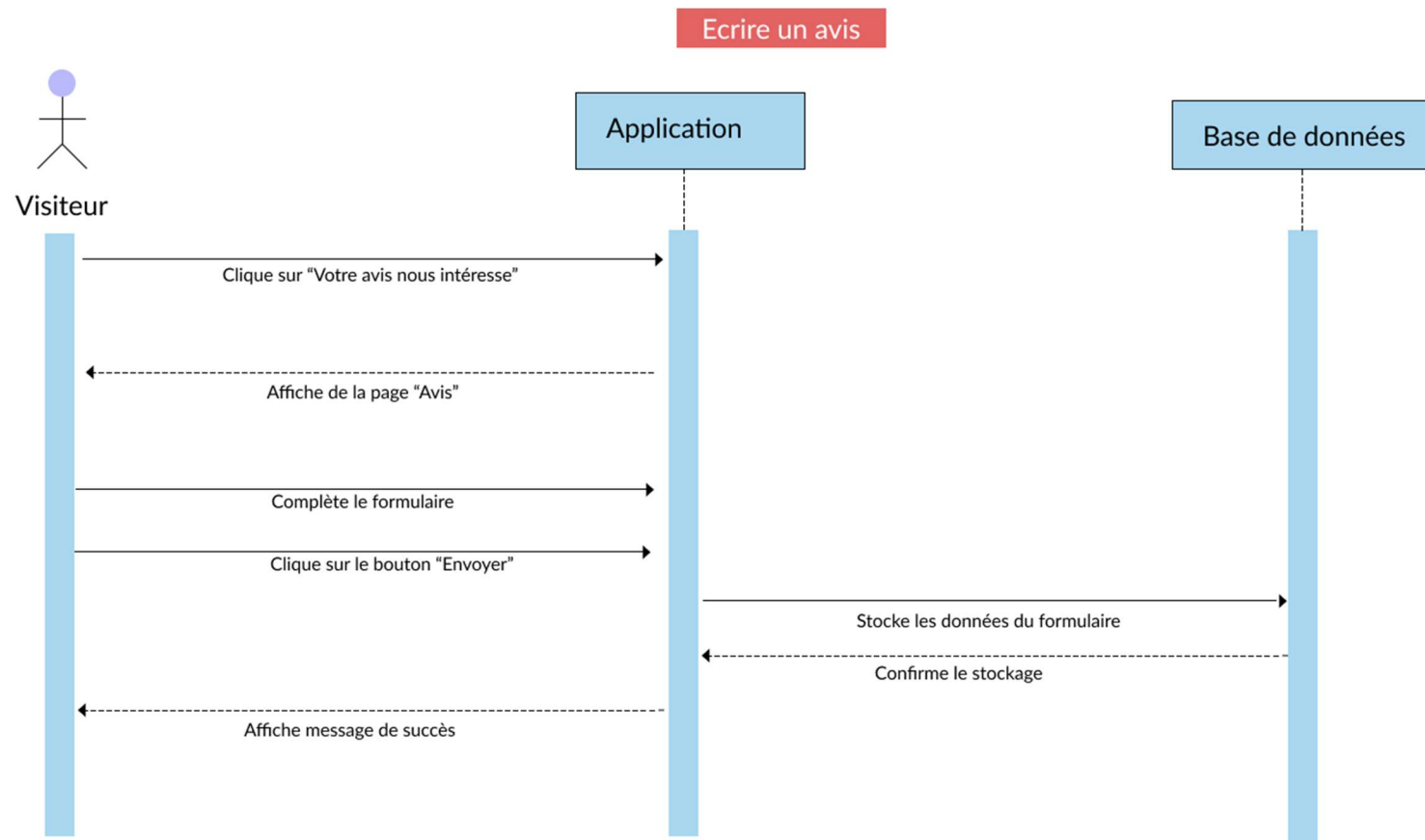
- Ajout de ClearDB MySQL sur Heroku
- Ajout de l'identifiant et mot de passe dans le fichier .env

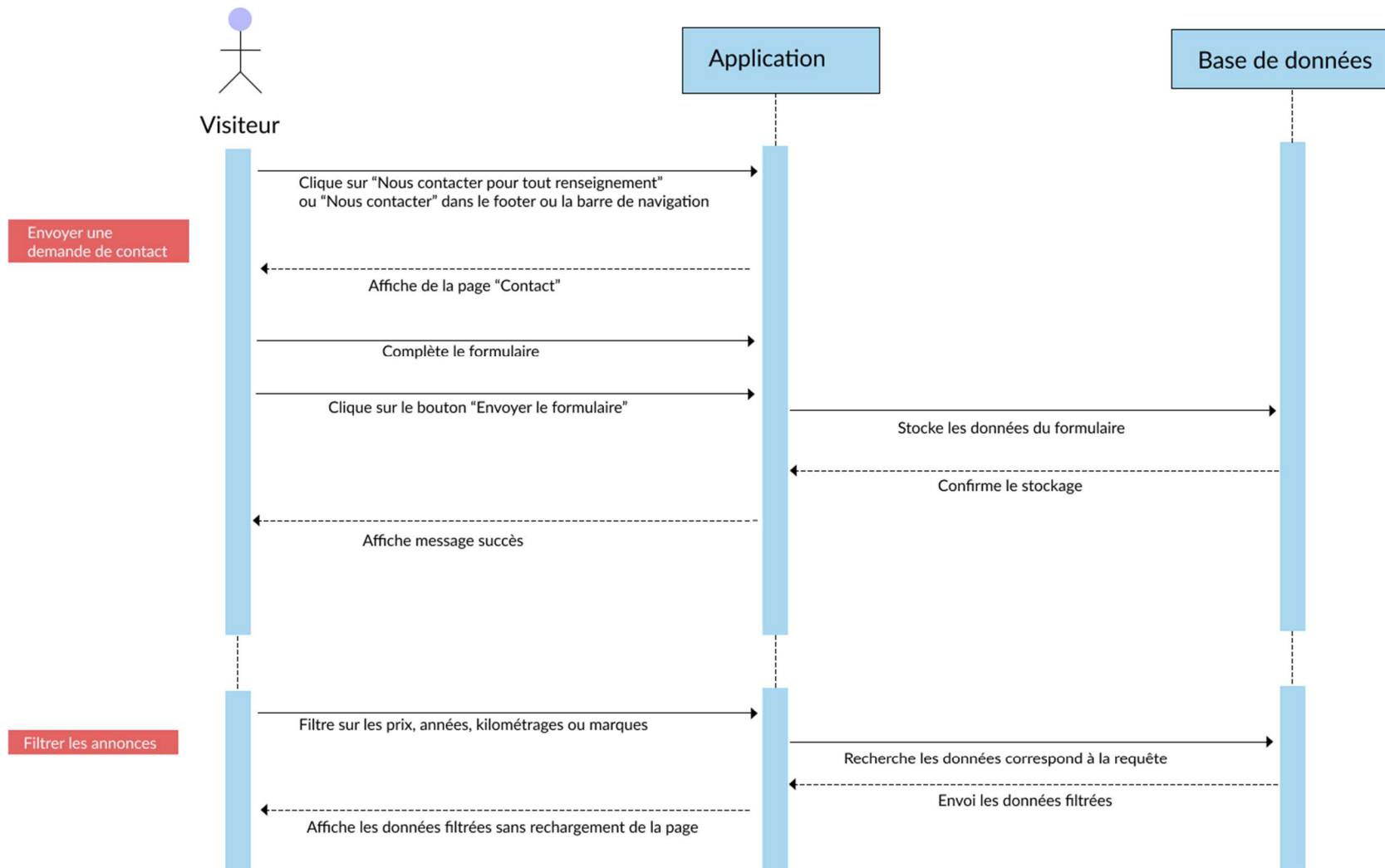
B. Diagramme de Cas d'utilisation

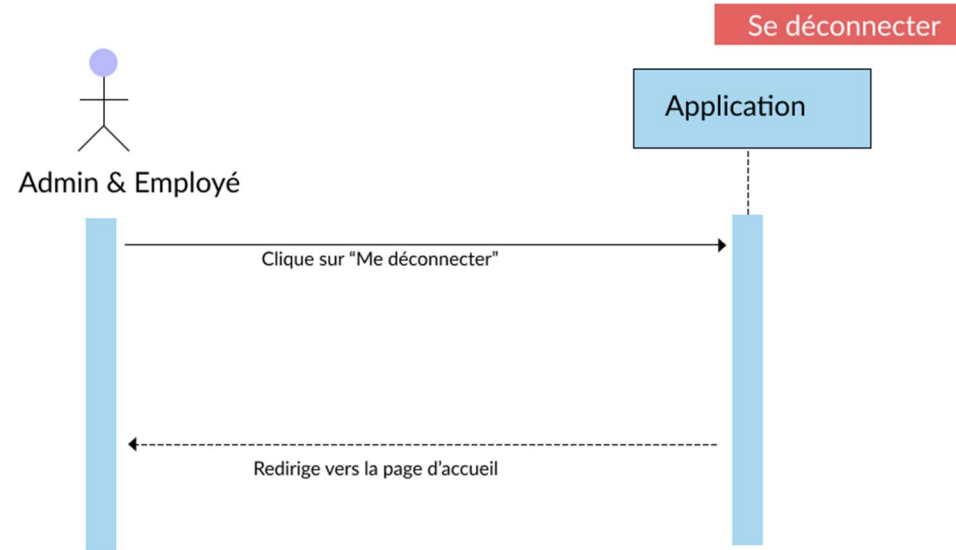
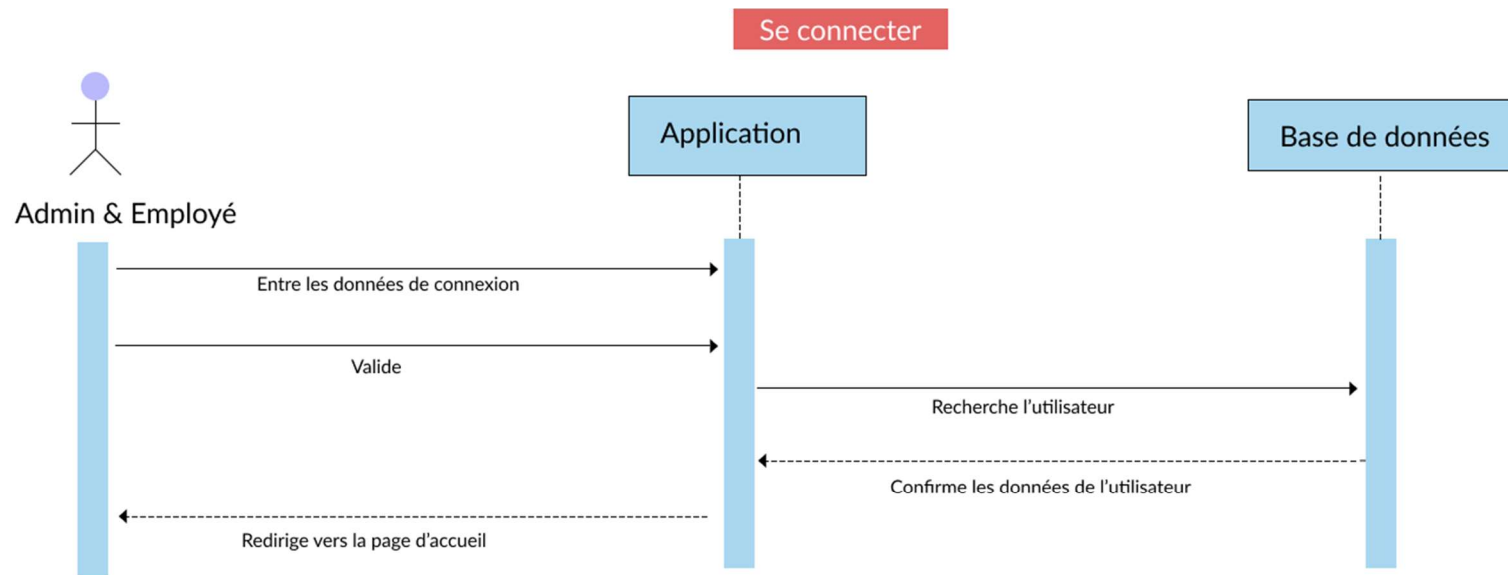


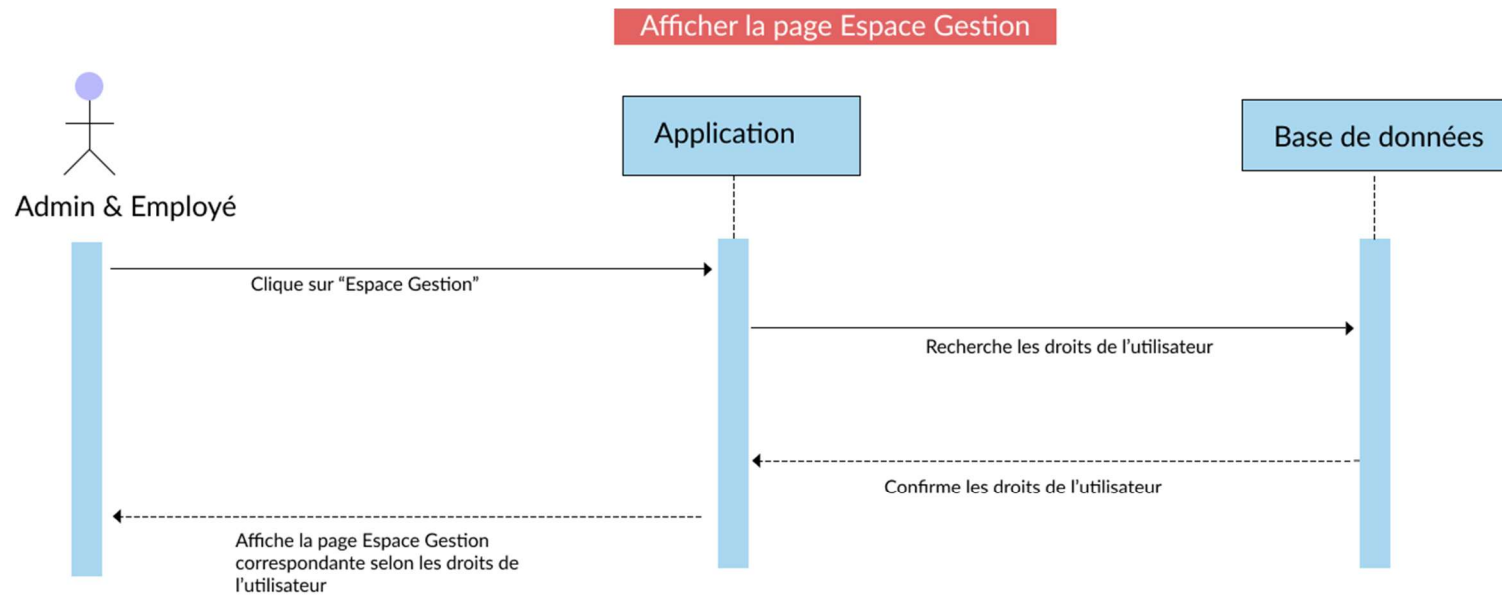
C. Diagramme de séquence

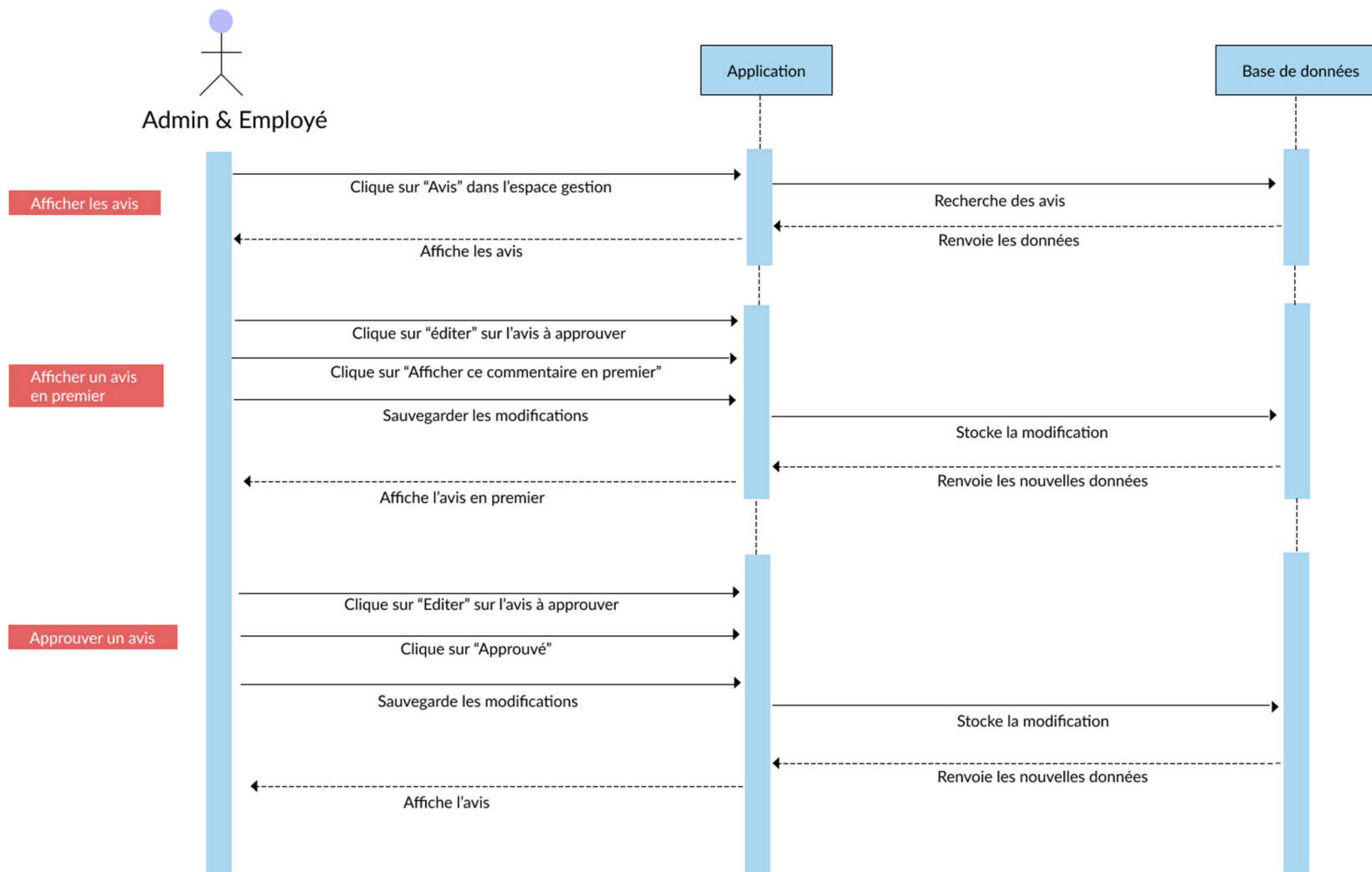






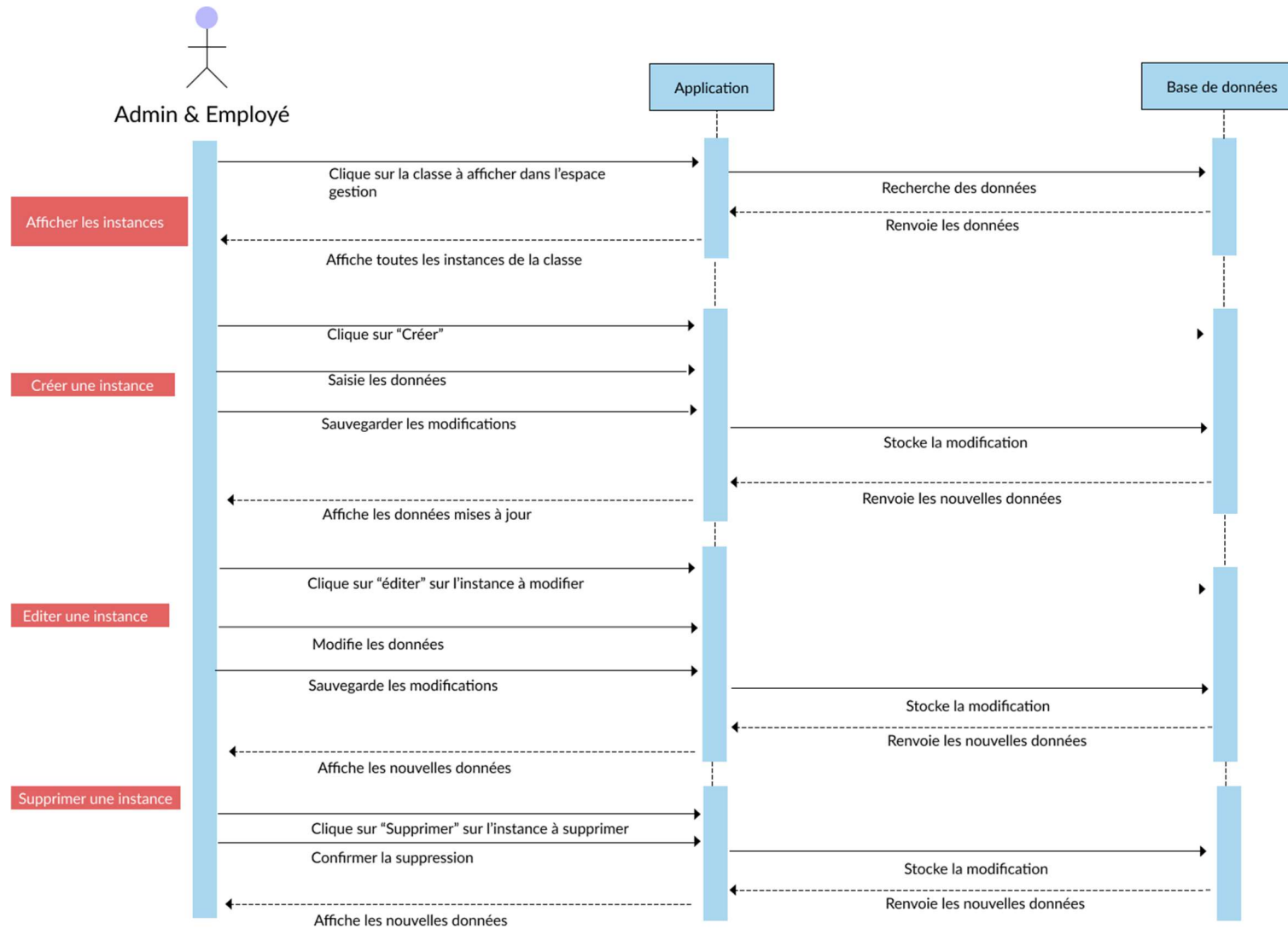






Pour l'employé, les classes modifiables correspondent à : une annonce, un avis, un contact ou une marque.

Pour l'admin, il s'agit des classes de l'employé auxquelles on ajoute : un service, les horaires d'ouverture, une annonce spéciale et un utilisateur.



4. Diagramme de classe

