

## **Description générale**

L'application est une plateforme de gestion de séjours hospitaliers destinée au personnel médical. Elle permet de visualiser, gérer et valider les séjours des patients dans différents services hospitaliers.

# Sommaire

<b>Description générale</b>	<b>1</b>
<b>Sommaire</b>	<b>2</b>
<b>Technologies utilisées</b>	<b>3</b>
<b>Architecture</b>	<b>3</b>
<b>Structure du projet</b>	<b>3</b>
<b>Fonctionnalités principales</b>	<b>4</b>
<b>Routes principales et fonctionnement des contrôleurs</b>	<b>4</b>
PrincipaleController	4
GestionSejourController	4
SejourController	5
UserController	5
SecurityController	5

## Technologies utilisées

- **Framework PHP** : Symfony 5
- **Base de données** : MySQL
- **Langages** : PHP, Twig (pour les templates HTML), HTML, CSS
- **ORM** : Doctrine
- **Pagination** : KnpPaginatorBundle
- **Système de routage** : Symfony Routing Component
- **Authentification** : Symfony Security Component

## Architecture

L'architecture de l'application suit le modèle MVC (Modèle-Vue-Contrôleur) :

- **Modèles (entités)** : Les entités sont des classes PHP qui représentent les données manipulées par l'application. Elles sont gérées par Doctrine ORM.
- **Vues** : Les vues sont des fichiers Twig qui définissent l'interface utilisateur de l'application.
- **Contrôleurs** : Les contrôleurs sont des classes PHP qui orchestrent les interactions entre les modèles et les vues. Ils contiennent la logique métier de l'application.

## Structure du projet

L'application est structurée selon les standards de Symfony :

- **config** : Contient les fichiers de configuration de l'application.
- **public** : Contient les fichiers publics accessibles depuis le navigateur (index.php, fichiers CSS, images).
- **src** : Contient le code source de l'application.
  - **Controller** : Contient les contrôleurs de l'application.
  - **Entity** : Contient les entités (modèles) de l'application.
  - **Form** : Contient les classes de formulaire.
  - **Repository** : Contient les classes Repository pour l'accès aux données.
- **templates** : Contient les fichiers Twig des vues de l'application.
- **tests** : Contient les tests unitaires et fonctionnels de l'application.

## Fonctionnalités principales

1. **Gestion des séjours :**
  - Affichage des séjours en cours dans un service spécifique.
  - Validation de la sortie d'un patient.
  - Recherche des séjours à une date donnée.
  - Affichage des séjours à venir dans un service.
2. **Gestion des utilisateurs :**
  - Affichage de la liste des utilisateurs.
  - Ajout, modification et suppression d'utilisateurs.
3. **Authentification :**
  - Connexion des utilisateurs via un formulaire de login.
  - Déconnexion des utilisateurs.

## Routes principales et fonctionnement des contrôleurs

### PrincipaleController

- **/ (Page d'accueil) :**
  - Contrôleur : PrincipaleController
  - Fonction : index()
  - Description : Affiche la page d'accueil de l'application, incluant la date actuelle.

### GestionSejourController

- **/sejour/sorties (Liste des séjours en cours) :**
  - Contrôleur : GestionSejourController
  - Fonction : sorties()
  - Description : Affiche la liste des séjours en cours dans le service de l'utilisateur connecté.
- **/sejour/sortie/{id} (Validation de la sortie) :**
  - Contrôleur : GestionSejourController
  - Fonction : valid()
  - Description : Permet de valider la sortie d'un patient d'un séjour en cours.
- **/sejour/date (Recherche des séjours par date) :**
  - Contrôleur : GestionSejourController
  - Fonction : gestionDateDonnee()
  - Description : Permet de rechercher les séjours à une date donnée dans le service de l'utilisateur connecté, la date du jour est la date donnée par défaut.
- **/sejour/aVenir (Affichage des séjours à venir) :**
  - Contrôleur : GestionSejourController
  - Fonction : gestionSejourAVenir()
  - Description : Affiche les séjours à venir dans le service de l'utilisateur connecté.

## SejourController

- **/sejour/{id} (Affichage détaillé d'un séjour) :**
  - Contrôleur : SejourController
  - Fonction : show()
  - Description : Affiche les détails d'un séjour spécifique si l'utilisateur est du même service que le séjour.

## UserController

- **/user (Liste des utilisateurs) :**
  - Contrôleur : UserController
  - Fonction : index()
  - Description : Affiche la liste de tous les utilisateurs.
- **/user/new (Ajout d'un nouvel utilisateur) :**
  - Contrôleur : UserController
  - Fonction : new()
  - Description : Permet d'ajouter un nouvel utilisateur.
- **/user/{id} (Affichage détaillé d'un utilisateur) :**
  - Contrôleur : UserController
  - Fonction : show()
  - Description : Affiche les détails d'un utilisateur spécifique.
- **/user/{id}/edit (Modification d'un utilisateur) :**
  - Contrôleur : UserController
  - Fonction : edit()
  - Description : Permet de modifier les informations d'un utilisateur.
- **/user/{id} (Suppression d'un utilisateur) :**
  - Contrôleur : UserController
  - Fonction : delete()
  - Description : Supprime un utilisateur spécifique.

## SecurityController

- **/login (Page de connexion) :**
  - Contrôleur : SecurityController
  - Fonction : login()
  - Description : Affiche le formulaire de connexion et gère le processus d'authentification.
- **/logout (Déconnexion de l'utilisateur) :**
  - Contrôleur : SecurityController
  - Fonction : logout()
  - Description : Gère la déconnexion de l'utilisateur.