RAPPORT PROFESSIONNEL

Application Web de Gestion de Rendez-vous Médicaux

SantéPlus

□□ TABLE DES MATIÈRES

□□ RÉSUMÉ EXÉCUTIF

SantéPlus est une application web moderne développée avec le framework Symfony 6.4, destinée à la gestion de rendez-vous médicaux en ligne. Cette plateforme permet aux patients de prendre rendez-vous avec des médecins, de consulter leurs rendez-vous à venir, et offre aux administrateurs un tableau de bord complet pour gérer l'ensemble du système.

Date du rapport : Janvier 2025
Version de l'application : 1.0
Framework principal : Symfony 6.4
Langage de programmation : PHP 8.1+

□□ PRÉSENTATION DU PROJET

Objectif

Développer une solution web complète et sécurisée pour faciliter la prise de rendez-vous médicaux en ligne, améliorer l'expérience patient et optimiser la gestion administrative des consultations médicales.

Portée du Projet

- **Public cible** : Patients, Médecins, Administrateurs
- **Domaine d'application** : Santé et bien-être
- **Type de solution** : Application web full-stack

□□ ARCHITECTURE TECHNIQUE

Architecture MVC (Model-View-Controller)

L'application suit l'architecture MVC standard de Symfony :

```
santeplus/
??? config/              # Configuration de l'application
??? migrations/          # Migrations de base de données
??? public/              # Point d'entrée public (assets, CSS, JS)
??? src/
?   ??? Command/         # Commandes console personnalisées
?   ??? Controller/      # Contrôleurs (logique métier)
?   ?   ??? Admin/       # Contrôleurs administration
?   ?   ??? ...          # Contrôleurs publics
?   ??? DataFixtures/    # Données de test
?   ??? Entity/          # Modèles de données (Doctrine ORM)
?   ??? Form/            # Formulaires Symfony
?   ??? Repository/      # Couche d'accès aux données
??? templates/           # Vues Twig
    ??? admin/           # Templates administration
    ??? home/            # Templates publics
    ??? ...
```

Pattern de Conception

- **Repository Pattern** : Accès aux données centralisé via les repositories Doctrine
- **Form Builder** : Gestion des formulaires via Symfony Forms
- **Dependency Injection** : Injection de dépendances via le conteneur Symfony

---

⚙ FONCTIONNALITÉS

1. Espace Public

Page d'Accueil (`/`)

- Présentation de la plateforme SantéPlus
- Section services (consultation en ligne et sur place)
- Vidéo promotionnelle intégrée
- Témoignages de patients

- Navigation intuitive vers les fonctionnalités principales

Liste des Médecins (`/medecins`)

- Affichage de tous les médecins disponibles
- Informations affichées : Nom, Spécialité, Ville, Téléphone
- Interface responsive et moderne

Prise de Rendez-vous (`/rendez-vous`)

- Formulaire de réservation accessible sans authentification
- Sélection du médecin, date et heure
- Gestion automatique des patients (création ou mise à jour)
- Vérification des doublons de rendez-vous
- Messages de confirmation et d'erreur
- Validation des données en temps réel

2. Espace Patient (Authentifié)

Dashboard Patient (`/dashboard`)

- Vue d'ensemble des rendez-vous du patient
- Affichage du prochain rendez-vous à venir
- Statistiques personnelles (nombre total de rendez-vous)
- Liste complète des rendez-vous (triés par date et heure)
- Accès rapide à la liste des médecins disponibles

Sécurité : Accès protégé par le rôle `ROLE_USER`

3. Espace Administration (`/admin`)

Tableau de Bord Administrateur (`/admin`)

- Statistiques globales :
  - Nombre total de patients
  - Nombre total de médecins
  - Rendez-vous du jour
  - Rendez-vous à venir
- Vue d'ensemble de l'activité de la plateforme

Gestion des Médecins (`/admin/medecin`)

- **CRUD complet** (Create, Read, Update, Delete)
- Création de nouveaux médecins
- Modification des informations (nom, spécialité, ville, téléphone)
- Suppression de médecins
- Consultation des détails

Gestion des Patients (`/admin/patient`)

- Consultation de tous les patients enregistrés
- Affichage des informations détaillées
- Modification des données patients
- Suppression de patients

Gestion des Rendez-vous (`/admin/rendezvous`)

- Liste complète de tous les rendez-vous
- Consultation des détails (patient, médecin, date, heure)
- Modification des rendez-vous
- Suppression de rendez-vous

□□ STRUCTURE DE LA BASE DE DONNÉES

Modèle de Données

L'application utilise **Doctrine ORM** pour la gestion de la persistance des données avec MySQL/MariaDB.

Entité : **Medecin**

```
- id (INT, Primary Key, Auto-increment)
- nom (VARCHAR 255)
- specialite (VARCHAR 255)
- ville (VARCHAR 255)
- telephone (VARCHAR 255)
- Relation : OneToMany avec RendezVous
```

Entité : **Patient**

```
- id (INT, Primary Key, Auto-increment)
- nom (VARCHAR 255)
- prenom (VARCHAR 255)
- email (VARCHAR 180, UNIQUE)
- adresse (VARCHAR 255)
- telephone (VARCHAR 255)
- password (VARCHAR 255) - Hashé avec Symfony PasswordHasher
- roles (JSON) - Système de rôles Symfony
- Relation : OneToMany avec RendezVous
```

Entité : **RendezVous**

```
- id (INT, Primary Key, Auto-increment)
- date (DATE)
- heure (TIME)
- medecin_id (INT, Foreign Key ? Medecin)
- patient_id (INT, Foreign Key ? Patient)
- Relations : ManyToOne avec Medecin et Patient
```

Relations

- **Medecin** ↔ **RendezVous** : Relation OneToMany
- **Patient** ↔ **RendezVous** : Relation OneToMany
- **RendezVous** : Table de liaison avec contraintes d'intégrité référentielle

Migrations

L'application utilise **Doctrine Migrations** pour la gestion des versions de schéma :

- `Version20260102210257.php` : Création initiale des tables
- `Version20260103082516.php` : Ajout des champs manquants (prenom, adresse, telephone pour Patient)
- `Version20260103131803.php` : Autres ajustements du schéma

□□ SÉCURITÉ

Authentification et Autorisation

Système de Rôles

- **ROLE_USER** : Accès à l'espace patient (dashboard)
- **ROLE_ADMIN** : Accès complet à l'administration (à implémenter)

Protection des Routes

- Routes publiques : Accueil, Liste médecins, Prise de rendez-vous
- Routes protégées : Dashboard patient (`#[IsGranted('ROLE_USER')]`)
- Routes admin : Toutes les routes `/admin/*` (protection à renforcer)

Sécurité des Données

- **Hachage des mots de passe** : Utilisation de `UserPasswordHasherInterface` de Symfony
- **Validation des formulaires** : Validation côté serveur via Symfony Validator
- **Protection CSRF** : Intégrée par défaut dans les formulaires Symfony
- **Prévention des doublons** : Vérification avant création de rendez-vous

Recommandations de Sécurité

- Implémenter la protection CSRF sur toutes les routes sensibles
- Ajouter un système de rate limiting pour les formulaires
- Mettre en place un système de logs pour les actions administratives
- Chiffrer les données sensibles en transit (HTTPS)

□□ INTERFACE UTILISATEUR

Design et Expérience Utilisateur

Caractéristiques

- **Design moderne et responsive** : Adaptation mobile, tablette et desktop
- **Mode sombre/clair** : Toggle de thème intégré
- **Navigation intuitive** : Menu de navigation clair et accessible
- **Messages flash** : Retours utilisateur pour les actions (succès, erreurs)
- **Logo et branding** : Identité visuelle SantéPlus

Assets

- **CSS personnalisé** : `public/css/style.css` et `public/css/admin.css`
- **JavaScript** : `public/js/script.js` pour les interactions dynamiques

- **Images** : Logo et images promotionnelles
- **Vidéos** : Vidéo promotionnelle intégrée (`medadom.mp4`)

Templates Twig

- **Template de base** : `base.html.twig` avec header et footer communs
- **Templates publics** : Home, Médecins, Rendez-vous
- **Templates admin** : Dashboard, CRUD pour chaque entité
- **Héritage de templates** : Réutilisation du code via l'héritage Twig

---

□□ TECHNOLOGIES UTILISÉES

Backend

- **PHP 8.1+** : Langage de programmation
- **Symfony 6.4** : Framework PHP moderne
  - Symfony Framework Bundle
  - Symfony Security Bundle
  - Symfony Form Component
  - Symfony Validator Component
  - Symfony Twig Bundle
  - Symfony Console Component

Base de Données

- **Doctrine ORM 3.6** : ORM pour la gestion de la persistance
- **Doctrine DBAL 3** : Abstraction de la base de données

- **Doctrine Migrations 3.7** : Gestion des versions de schéma
- **MySQL/MariaDB** : SGBD relationnel

Frontend

- **Twig 3.x** : Moteur de templates
- **HTML5 / CSS3** : Structure et style
- **JavaScript (Vanilla)** : Interactivité côté client

Outils de Développement

- **Composer** : Gestionnaire de dépendances PHP
- **Symfony Maker Bundle** : Génération de code
- **Doctrine Fixtures Bundle** : Données de test
- **Symfony Flex** : Installation et configuration automatique

Dépendances Principales

```
{
  "symfony/framework-bundle": "6.4.*",
  "symfony/security-bundle": "6.4.*",
  "symfony/form": "6.4.*",
  "symfony/validator": "6.4.*",
  "doctrine/orm": "^3.6",
  "doctrine/doctrine-bundle": "^2.18",
  "twig/twig-bundle": "6.4.*"
}
```

□ POINTS FORTS

1. Architecture Solide

- Respect des bonnes pratiques Symfony

- Séparation claire des responsabilités (MVC)
- Code modulaire et maintenable

2. Sécurité

- Système d'authentification intégré
- Hachage sécurisé des mots de passe
- Protection des routes sensibles

3. Expérience Utilisateur

- Interface intuitive et moderne
- Messages de retour clairs
- Design responsive

4. Gestion des Données

- ORM puissant (Doctrine)
- Migrations pour la gestion des versions
- Relations bien définies entre entités

5. Fonctionnalités Complètes

- CRUD complet pour toutes les entités
- Gestion des rendez-vous avec prévention des doublons
- Tableaux de bord informatifs

6. Extensibilité

- Architecture permettant l'ajout facile de nouvelles fonctionnalités
- Utilisation de services et d'injection de dépendances

□□ RECOMMANDATIONS ET AMÉLIORATIONS FUTURES

Court Terme

1. **Sécurité**
   - Implémenter le rôle `ROLE_ADMIN` avec protection complète des routes admin
   - Ajouter la validation des emails et téléphones
   - Mettre en place un système de réinitialisation de mot de passe

2. **Fonctionnalités**
   - Ajouter la possibilité d'annuler/modifier un rendez-vous depuis le dashboard patient
   - Implémenter un système de notifications (email/SMS) pour les rappels
   - Ajouter un calendrier visuel pour la sélection des dates

3. **Interface**
   - Améliorer l'accessibilité (WCAG 2.1)
   - Ajouter des filtres de recherche pour les médecins (par spécialité, ville)
   - Implémenter la pagination pour les listes longues

Moyen Terme

1. **Fonctionnalités Avancées**
   - Système de disponibilité des médecins (horaires de travail)

- Gestion des créneaux horaires disponibles
- Historique des consultations
- Notes et commentaires sur les rendez-vous

2. **Performance**
   - Mise en cache des requêtes fréquentes
   - Optimisation des requêtes Doctrine
   - Lazy loading pour les relations

3. **API REST**
   - Développer une API REST pour une application mobile future
   - Documentation API (OpenAPI/Swagger)

Long Terme

1. **Intégrations**
   - Intégration avec des systèmes de paiement en ligne
   - Connexion avec des systèmes de gestion hospitalière
   - Intégration avec des services de télémédecine

2. **Analytics**
   - Tableaux de bord analytiques avancés
   - Rapports statistiques (taux d'occupation, spécialités les plus demandées)
   - Export de données (CSV, PDF)

3. **Multi-tenant**

- Support de plusieurs établissements médicaux
- Gestion de plusieurs centres de santé

□□ MÉTRIQUES ET STATISTIQUES

Code

- **Nombre d'entités** : 3 (Medecin, Patient, RendezVous)
- **Nombre de contrôleurs** : 8 (4 publics + 4 admin)
- **Nombre de formulaires** : 4
- **Nombre de templates** : 20+

Base de Données

- **Tables** : 3 tables principales
- **Relations** : 2 relations ManyToOne
- **Migrations** : 3 versions

Fonctionnalités

- **Routes publiques** : 4
- **Routes authentifiées** : 1 (dashboard)
- **Routes admin** : 12+ (CRUD complet)

□□ CONCLUSION

SantéPlus est une application web robuste et bien structurée qui répond aux besoins de base d'une plateforme de gestion de rendez-vous médicaux. L'utilisation de Symfony 6.4 garantit une base solide, sécurisée et maintenable.

Points Clés

- Architecture moderne et scalable
- Sécurité intégrée
- Interface utilisateur intuitive
- Code maintenable et extensible
- Fonctionnalités complètes pour la gestion des rendez-vous

Prochaines Étapes Recommandées

1. Renforcer la sécurité (rôles admin, validation)
2. Améliorer l'expérience utilisateur (calendrier, notifications)
3. Optimiser les performances (cache, requêtes)
4. Préparer l'extensibilité (API REST)

L'application est prête pour un déploiement en environnement de développement et peut être étendue selon les besoins spécifiques de l'établissement médical.

INFORMATIONS DE CONTACT

Projet : SantéPlus
Type : Application Web de Gestion de Rendez-vous Médicaux
Framework : Symfony 6.4
Environnement : XAMPP (Windows)
Date de création : Janvier 2025

Rapport généré le : Janvier 2025

Version du document : 1.0