



SoigneMoi

Centre Hospitalier

Cahier des charges

Table des matières

1. Introduction	3
1.1. Objectif	3
1.2. Contexte	3
2. Description générale	3
2.1. Application Web Principale	3
2.2. Application Mobile pour les Médecins	3
2.3. Application Bureautique pour les Secrétaires	3
3. Spécifications fonctionnelles	4
3.1 Exigences utilisateur	4
3.2 Navigation	5
3.3 Flux de travail	5
4. Spécifications techniques	6
4.1 Technologies utilisées	6
4.2 Architecture	6
4.3 Sécurité	7
5. Interfaces utilisateur	8
5.1 Conception visuelle	8
5.2 Maquettes	8
6. Gestion des données	9
Structure de la base de données	9
7. Performances	10
7.1 Temps de chargement	10
7.2 Autres mesures de performance	10
8. Maintenance et évolution	11
8.1 Documentation	11
8.2 Maintenance	11
9. Contraintes	12
9.1 Contraintes techniques	12
9.2 Contraintes de délais	12
10. Annexes	13
Glossaire	13
Structure de la base de données	14

1. Introduction

1.1. Objectif

L'objectif de ce document est de définir les spécifications fonctionnelles et techniques des applications développées pour SoigneMoi. Ces applications comprennent une application web principale, une application mobile destinée aux médecins et une application bureautique pour les secrétaires de l'hôpital

1.2. Contexte

SoigneMoi est un hôpital situé dans la région lilloise, confronté à des problèmes d'efficacité dus à des processus manuels. Leur accueil et leur prise en charge des patients sont lents, en partie à cause du manque d'un agenda informatisé pour les praticiens.

L'Hospital a fait appel à notre entreprise dans le but d'améliorer l'efficacité de l'accueil des patients, de la prise en charge médicale et de la gestion administrative, en fournissant des outils numériques adaptés.

2. Description générale

2.1. Application Web Principale

L'application web principale, "SoigneMoi Web", est conçue pour permettre aux patients de réserver leurs séjours à l'hôpital, aux médecins de consulter leur emploi du temps et aux administrateurs de gérer les admissions et les sorties des patients. Elle offre également un historique des séjours pour les utilisateurs. Cette application doit être responsive, offrant ainsi une expérience utilisateur optimale sur les ordinateurs de bureau, les tablettes et les smartphones.

2.2. Application Mobile pour les Médecins

"SoigneMoi Mobile" est une application mobile destinée aux médecins, leur permettant de saisir des prescriptions médicales et de donner des avis sur l'état des patients. Ils peuvent consulter les dossiers médicaux des patients et recevoir des alertes concernant les nouvelles tâches.

2.3. Application Bureautique pour les Secrétaires

"SoigneMoi Office" est une application bureautique destinée aux secrétaires de l'hôpital. Elle leur permet de visualiser les entrées et sorties des patients, de gérer les dossiers médicaux et de générer des rapports sur l'activité de l'hôpital.

3. Spécifications fonctionnelles

3.1 Exigences utilisateur

Les exigences des utilisateurs pour l'application "SoigneMoi" sont définies en fonction de leurs rôles et de leurs besoins spécifiques. Voici une description des principaux rôles et des actions possibles pour chaque persona :

Personnas :

1. Visiteur :

- **Objectif** : Trouver des informations sur les services offerts par l'hôpital et réserver un séjour si nécessaire.
- **Actions possibles** :
 - Créer un compte utilisateur.
 - Consulter les informations sur l'hôpital et ses services.
 - Réserver un séjour en spécifiant les dates et le motif de visite.
 - Consulter l'historique des séjours précédents (après avoir créé un compte).

2. Patient :

- **Objectif** : Réserver et gérer ses séjours à l'hôpital, ainsi que consulter son historique médical.
- **Actions possibles** :
 - Se connecter à son compte utilisateur.
 - Consulter son historique de séjours.
 - Réserver un nouveau séjour en spécifiant les détails nécessaires.
 - Consulter les détails de son séjour en cours ou à venir.

3. Médecin :

- **Objectif** : Gérer son emploi du temps, consulter les informations des patients et saisir des données médicales.
- **Actions possibles** :
 - Consulter son emploi du temps et ses disponibilités.
 - Accéder aux dossiers médicaux des patients et saisir des données médicales.
 - Donner des avis sur l'état des patients et prescrire des médicaments.

4. Administrateur :

- **Objectif** : Gérer les comptes utilisateurs, les informations des médecins et les plannings.
- **Actions possibles** :
 - Gérer les comptes utilisateurs (ajouter, modifier, supprimer).
 - Ajouter, modifier ou supprimer des informations sur les médecins.
 - Gérer les plannings des médecins (ajouter, modifier les rendez-vous).

3.2 Navigation

La structure de navigation de l'application "SoigneMoi" est conçue pour faciliter l'accès aux fonctionnalités principales pour chaque persona. Voici une description de la navigation pour chaque persona :

- **Visiteur :**
 - Accueil
 - À propos
 - Services
 - Contact
 - Connexion/Inscription
- **Patient :**
 - Tableau de bord (historique des séjours)
 - Réserver un séjour
 - Consulter les informations médicales
- **Médecin/Praticien :**
 - Tableau de bord (emploi du temps)
 - Patients (consultation des dossiers médicaux)
 - Prescription médicale
 - Donner un avis sur l'état des patients
- **Administrateur :**
 - Gérer les comptes utilisateurs
 - Gérer les informations des médecins/praticiens
 - Gérer les plannings

3.3 Flux de travail

Les flux de travail dans l'application "SoigneMoi" sont adaptés aux besoins spécifiques de chaque persona et suivent le processus typique d'un séjour à l'hôpital. Voici une description des principaux flux de travail :

1. **Réservation d'un séjour :**
 - Le visiteur (patient) sélectionne les dates de séjour et le motif de sa visite.
 - Il choisit la spécialité médicale et le médecin désiré (si applicable).
 - Après confirmation, le séjour est enregistré dans le système.
2. **Gestion des séjours :**
 - Le patient accède à son tableau de bord pour consulter les détails de son séjour en cours ou à venir.
 - Il peut également consulter son historique de séjours pour accéder à ses informations médicales passées.
3. **Consultation médicale :**
 - Le médecin accède à son tableau de bord pour consulter son emploi du temps et ses rendez-vous.

- Il peut consulter les dossiers médicaux des patients, saisir des données médicales, prescrire des médicaments et donner des avis sur leur état.
4. **Gestion administrative :**
- L'administrateur accède à l'interface d'administration pour gérer les comptes utilisateurs, les informations des médecins et les plannings.

4. Spécifications techniques

4.1 Technologies utilisées

Pour les différentes applications de "SoigneMoi", les technologies suivantes seront utilisées :

- **Application Web Principale (Symfony) :**
 - Langage de programmation : PHP
 - Framework : Symfony
 - Bibliothèques front-end : Bootstrap
 - Base de données : MySQL
- **Application Mobile (Flutter) :**
 - Langage de programmation : Dart
 - Framework : Flutter
 - Communication avec le back-end via des requêtes HTTP
- **Application Bureautique (Python) :**
 - Langage de programmation : Python
 - Bibliothèques : Django (pour le back-end)
 - Communication avec le back-end via des requêtes HTTP ou des appels de fonction

4.2 Architecture

- **Application Web Principale (Symfony) :**
 - Utilisation du modèle MVC (Modèle-Vue-Contrôleur) avec Symfony.
 - Modèle : Doctrine pour la gestion des données.
 - Vue : Utilisation de templates Twig.
 - Contrôleur : Gestion des requêtes et des réponses.
- **Application Mobile (Flutter) :**
 - Utilisation de l'architecture MVVM (Modèle-Vue-ViewModèle) avec Flutter.
 - Modèle : Gestion des données avec des objets Dart.
 - Vue : Création d'interfaces utilisateur réactives avec des widgets Flutter.
 - ViewModèle : Gestion de la logique métier et des états.
- **Application Bureautique (Python) :**
 - Architecture basée sur les principes de développement web en Python.
 - Utilisation de Flask ou Django pour le back-end.
 - Communication avec le front-end via des API RESTful ou des appels de fonction directe.

4.3 Sécurité

Pour garantir la protection des données des patients et assurer la conformité au RGPD (Règlement Général sur la Protection des Données), des mesures de sécurité seront mises en place tout au long du développement :

- **Application Web Principale (Symfony) :**
 - Utilisation des composants de sécurité intégrés à Symfony pour gérer l'authentification, les rôles et les autorisations des utilisateurs.
 - Chiffrement des données sensibles stockées dans la base de données.
 - Protection contre les attaques par injection SQL et XSS pour garantir l'intégrité des données des patients.
 - Conformité au RGPD assurée par des pratiques de sécurité robustes.
- **Application Mobile (Flutter) :**
 - Sécurisation des communications avec le back-end via HTTPS pour protéger la confidentialité des données en transit.
 - Gestion sécurisée des informations d'authentification et des jetons d'accès.
 - Cryptage des données sensibles stockées localement sur l'appareil pour prévenir tout accès non autorisé.
 - Conformité au RGPD assurée par la protection des données des patients et le respect de leur confidentialité.
- **Application Bureautique (Python) :**
 - Implémentation de bonnes pratiques de sécurité en Python, notamment la validation des entrées utilisateur.
 - Gestion sécurisée des sessions et des autorisations pour contrôler l'accès aux données sensibles.
 - Sécurisation des données transitant entre le front-end et le back-end pour garantir leur confidentialité.
 - Conformité au RGPD garantie par des mesures de sécurité rigoureuses et une protection renforcée des données des patients.

En mettant en place ces mesures de sécurité à chaque niveau d'application, SoigneMoi assure la confidentialité et l'intégrité des données des patients, ainsi que leur conformité aux réglementations en vigueur.

5. Interfaces utilisateur

5.1 Conception visuelle

L'application "SoigneMoi" adoptera une conception visuelle moderne et conviviale, avec une attention particulière portée à l'expérience utilisateur. Voici les principes de conception visuelle qui guideront le développement de l'interface utilisateur :

- **Couleurs** : Une palette de couleurs apaisantes et professionnelles sera utilisée, avec des tons de bleu et de blanc pour évoquer la confiance, la propreté et la tranquillité. Des nuances de vert pourraient également être utilisées pour rappeler l'environnement médical.
- **Polices** : Des polices lisibles et sans-serif seront privilégiées pour assurer une bonne lisibilité sur tous les appareils. Des polices comme Roboto ou Open Sans seront utilisées pour leur clarté et leur polyvalence.
- **Disposition** : L'interface utilisateur sera conçue de manière à être intuitive et facile à naviguer. Les éléments seront disposés de manière logique, avec une attention particulière portée à l'accessibilité et à l'ergonomie.
- **Contraste** : Un contraste adéquat sera maintenu entre les éléments de texte et leur arrière-plan pour garantir une lisibilité optimale, en particulier pour les utilisateurs ayant des difficultés de vision.
- **Iconographie** : Des icônes claires et significatives seront utilisées pour aider les utilisateurs à comprendre rapidement les actions disponibles. Les icônes seront cohérentes et conformes aux normes de conception modernes.

5.2 Maquettes

Voici quelques maquettes illustrant le design de l'interface utilisateur pour différentes parties de l'application "SoigneMoi" :

<https://github.com/Maryonete/chu/tree/main/documentation/wireframes>

<https://github.com/Maryonete/chu/tree/main/documentation/mockup>

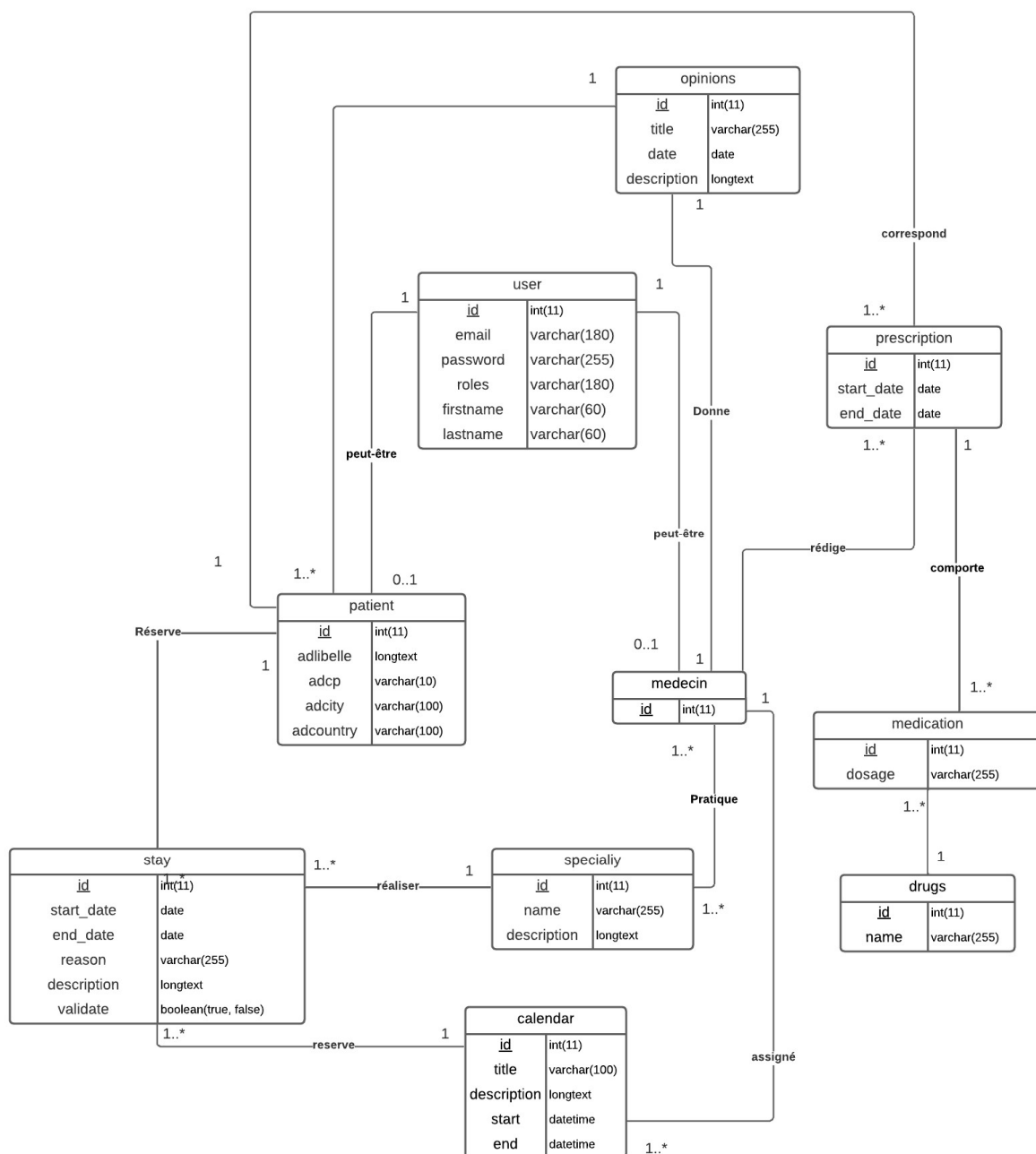
Ces maquettes sont des exemples illustrant la conception visuelle de l'application. Elles seront affinées et adaptées tout au long du processus de développement pour répondre aux besoins spécifiques des utilisateurs et garantir une expérience utilisateur optimale.

6. Gestion des données

Le **Modèle Conceptuel de Données (MCD)** est une représentation visuelle des entités, des relations et des attributs dans une base de données. Il aide à comprendre les besoins métier et à concevoir une base de données qui reflète précisément ces exigences.

Structure de la base de données

MCD - Modèle conceptuel des Données



Vous trouverez en annexe le détail des tables

7. Performances

7.1 Temps de chargement

Les exigences de performance en termes de temps de chargement des pages pour l'application "SoigneMoi" sont définies comme suit :

- **Temps de chargement optimal** : Le temps de chargement des pages de l'application doit être aussi court que possible pour offrir une expérience utilisateur fluide et réactive.
- **Objectif de chargement rapide** : L'objectif est de garantir que les pages se chargent en moins de X secondes, où X représente le temps cible défini pour chaque type de page (par exemple, la page d'accueil, la page de liste des rendez-vous, la page de détails du patient, etc.).
- **Mesure et optimisation continue** : Les temps de chargement seront mesurés régulièrement à l'aide d'outils de surveillance de la performance web. En cas de dépassement des seuils définis, des mesures d'optimisation seront mises en œuvre pour améliorer les performances.
- **Utilisation de bonnes pratiques de développement** : Le code source de l'application sera optimisé en utilisant des techniques telles que la minification des fichiers CSS et JavaScript, la compression des images, la mise en cache efficace des ressources statiques, et d'autres meilleures pratiques de développement web.
- **Tests de charge et de performance** : Des tests de charge et de performance seront effectués pour évaluer la capacité de l'application à gérer un volume élevé de demandes et à maintenir des temps de réponse rapides même dans des conditions de charge élevée.

7.2 Autres mesures de performance

Aucune mesure de performance supplémentaire n'a été spécifiée pour le moment. Les performances seront continuellement évaluées et des ajustements seront apportés au besoin pour garantir une expérience utilisateur optimale.

En respectant ces exigences de performance en termes de temps de chargement, l'application "SoigneMoi" garantit une expérience utilisateur fluide et efficace, contribuant ainsi à la satisfaction des utilisateurs et à l'atteinte des objectifs métier.

8. Maintenance et évolution

8.1 Documentation

- **Documentation technique** : Une documentation technique complète sera fournie, couvrant l'architecture de l'application, la structure de la base de données, les technologies utilisées, les API disponibles, ainsi que les instructions pour le déploiement et la configuration de l'application.
- **Manuel d'utilisation** : Un manuel d'utilisation détaillé sera fourni pour les utilisateurs finaux, expliquant les fonctionnalités de l'application, les flux de travail, et fournissant des instructions étape par étape sur la façon d'utiliser chaque fonctionnalité.
- **Guide de maintenance** : Un guide de maintenance sera élaboré pour l'équipe de développement, décrivant les procédures de maintenance régulières, les meilleures pratiques de gestion des mises à jour, les processus de sauvegarde des données, et d'autres tâches liées à la maintenance de l'application.

8.2 Maintenance

- **Mises à jour régulières** : L'application bénéficiera de mises à jour régulières pour corriger les bugs, améliorer les performances, et ajouter de nouvelles fonctionnalités en réponse aux besoins émergents des utilisateurs et aux évolutions du marché.
- **Processus de correction des bugs** : Un processus formel de gestion des incidents sera mis en place pour signaler, suivre et corriger les bugs rapidement. Les bugs critiques seront traités en priorité et des correctifs seront déployés dès que possible.
- **Suivi des performances** : Les performances de l'application seront surveillées en continu à l'aide d'outils de surveillance de la performance web. Des ajustements seront apportés au code et à l'infrastructure en fonction des résultats de ces analyses pour maintenir des performances optimales.
- **Gestion des versions** : Un système de contrôle des versions sera utilisé pour gérer le code source de l'application. Les mises à jour seront déployées de manière contrôlée, en tenant compte de l'impact sur la stabilité et la fonctionnalité de l'application.
- **Communication avec les utilisateurs** : Les utilisateurs seront informés des mises à jour importantes, des correctifs de sécurité et des nouveautés de l'application via des communications claires et transparentes, telles que des notifications dans l'application, des e-mails ou des annonces sur le site web.

En mettant en œuvre ces modalités de maintenance et d'évolution, l'application "SoigneMoi" restera robuste, sécurisée et à jour, répondant ainsi aux besoins changeants des utilisateurs et assurant une expérience utilisateur de haute qualité sur le long terme.

9. Contraintes

9.1 Contraintes techniques

- **Compatibilité avec les navigateurs** : L'application doit être compatible avec les principaux navigateurs web tels que Google Chrome, Mozilla Firefox, Safari et Microsoft Edge, ainsi qu'avec différentes résolutions d'écran pour garantir une expérience utilisateur cohérente.
- **Sécurité des données** : La confidentialité et la sécurité des données des patients doivent être garanties conformément aux normes de sécurité et de protection des données en vigueur, telles que le RGPD (Règlement général sur la protection des données) en Europe.
- **Performance** : L'application doit être optimisée pour des temps de chargement rapides et une réactivité élevée, même lorsqu'elle est soumise à une charge importante.
- **Scalabilité** : L'architecture de l'application doit être conçue de manière à pouvoir évoluer et à supporter une augmentation du nombre d'utilisateurs et de données à l'avenir.
- **Intégration des technologies** : L'application doit intégrer les technologies spécifiées pour chaque composant (Symfony pour l'application web principale, Flutter pour l'application mobile, Python pour l'application bureautique) et utiliser des normes de développement compatibles avec ces frameworks.

9.2 Contraintes de délais

- **Délai de livraison initial** : Le développement de l'application doit être réalisé dans le cadre d'un planning défini, avec des jalons clairement définis pour chaque phase du projet.
- **Dates limites à respecter** : Les différentes phases du projet, y compris la conception, le développement, les tests et le déploiement, doivent respecter les échéances convenues avec le client. Par exemple :
 - **Conception initiale** : À compléter d'ici le 15 mars 2024.
 - **Développement de l'application web** : À finaliser d'ici le 30 avril 2024.
 - Développement de l'application mobile : À terminer d'ici le 15 juin 2024.
 - Développement de l'application bureautique : À achever d'ici le 30 juillet 2024.
 - **Tests et validation** : Doivent être achevés d'ici le 15 septembre 2024.
 - **Déploiement et mise en production** : Prévu pour le 1er octobre 2024.
- **Communication et coordination** : Une communication étroite avec le client est essentielle pour s'assurer que les délais de livraison sont compris et respectés. Tout retard potentiel doit être anticipé et communiqué rapidement pour permettre une gestion efficace des attentes et des priorités.

En tenant compte de ces contraintes de délais, le projet "SoigneMoi" sera planifié et exécuté de manière efficace, en veillant à respecter les échéances fixées et à répondre aux besoins du client dans les délais impartis.

10. Annexes

Glossaire

- **Application Bureautique (Python)** : L'application bureautique développée en Python. Destinée aux secrétaires de l'hôpital, elle permet la gestion des admissions, des entrées et sorties des patients, ainsi que l'accès aux détails des dossiers patients.
- **Application Mobile (Flutter)** : L'application mobile développée avec le framework Flutter. Destinée aux médecins, elle permet la saisie des prescriptions, la consultation des avis médicaux, et d'autres fonctionnalités liées à la prise en charge des patients.
- **Application Web Principale (Symfony)** : L'application web principale développée avec le framework Symfony. Elle permet aux utilisateurs d'accéder aux fonctionnalités de base de SoigneMoi, telles que la création de compte, la gestion des séjours, et la visualisation des historiques.
- **Flutter** : Un framework open-source développé par Google pour la création d'applications multiplateformes, notamment pour les plateformes mobiles telles qu'Android et iOS. Il permet le développement d'interfaces utilisateur réactives et est particulièrement apprécié pour sa performance et sa productivité.
- **Framework Symfony** : Un framework PHP open-source utilisé pour le développement d'applications web. Il fournit une architecture modulaire, des composants réutilisables et des outils intégrés pour accélérer le processus de développement et garantir la robustesse et la sécurité des applications.
- **GitHub** : Une plateforme de développement logiciel basée sur Git, un système de contrôle de version. GitHub permet aux développeurs de collaborer, de partager et de travailler sur des projets de logiciel en ligne, en fournissant des fonctionnalités telles que le suivi des problèmes, les demandes de tirage (pull requests), et le déploiement continu.
- **PHPMyAdmin** : Un outil de gestion de base de données MySQL en ligne, qui offre une interface utilisateur graphique pour effectuer des opérations telles que la création de tables, l'exécution de requêtes SQL, et la gestion des utilisateurs et des privilèges.
- **PHPStan** : Un outil d'analyse statique pour le langage de programmation PHP. PHPStan analyse le code source PHP pour détecter les erreurs potentielles, les incohérences et les mauvaises pratiques de programmation, fournissant ainsi des recommandations pour améliorer la qualité et la fiabilité du code.
- **PHPUnit** : Un framework de test unitaire pour le langage de programmation PHP. Il permet aux développeurs d'écrire et d'exécuter des tests automatisés pour vérifier le bon fonctionnement de leurs applications PHP.
- **PHPUnit Security Advisories** : Un outil fourni par PHPUnit pour vérifier les vulnérabilités de sécurité connues dans les versions des dépendances utilisées par un projet. Il permet aux développeurs de détecter et de corriger les failles de sécurité potentielles dans leurs applications.
- **Python** : Un langage de programmation de haut niveau, polyvalent et facile à apprendre, largement utilisé dans le développement web, scientifique, et pour l'automatisation de tâches diverses.
- **Responsive Design** : Une approche de conception web qui vise à offrir une expérience utilisateur optimale sur différents appareils et tailles d'écran, en ajustant dynamiquement la mise en page et les éléments visuels pour s'adapter à chaque résolution.
- **RGPD (Règlement Général sur la Protection des Données)** : Un règlement de l'Union européenne qui vise à renforcer et à unifier la protection des données des individus au sein

de l'UE. Il impose des obligations strictes aux entreprises qui collectent, stockent et traitent les données personnelles des utilisateurs.

- **Symfony** : Un framework PHP moderne et populaire utilisé pour le développement d'applications web robustes, évolutives et sécurisées. Il suit le modèle de conception MVC (Modèle-Vue-Contrôleur) et propose une grande variété de composants prêts à l'emploi pour accélérer le développement.
- **Symfony Panther** : Une bibliothèque PHP basée sur Symfony pour effectuer des tests fonctionnels automatisés sur des applications web. Panther utilise un navigateur réel (comme Chrome ou Firefox) pour simuler les interactions utilisateur et tester les fonctionnalités de l'application.

Structure de la base de données

La gestion des données de l'application "SoigneMoi" est assurée par la structure de base de données relationnelle présentée ci-dessous, détaillant les tables, leurs champs pour assurer l'intégrité des données :

- Table **calendar** : Contient les événements planifiés associés aux médecins, avec des champs comme l'identifiant de médecin (**medecin_id**), l'identifiant de séjour (**stay_id**), le titre de l'événement (**title**), les dates de début (**start**) et de fin (**end**) ainsi qu'une description optionnelle.
- Table **drugs** : Liste des médicaments disponibles dans le système, enregistrés avec un nom unique (**name**).
- Table **medecin** : Table des médecins enregistrés, liés à un utilisateur via **user_id**.
- Table **medecin_speciality** : Associe les médecins à leurs spécialités respectives grâce à des clés étrangères vers les tables **medecin** et **speciality**.
- Table **medication** : Enregistre les prescriptions de médicaments pour les patients, incluant l'identifiant du médicament (**drug_id**), de la prescription (**prescription_id**) et le dosage (**dosage**).
- Table **opinions** : Contient les avis des médecins sur les patients, avec des champs comme l'identifiant du patient (**patient_id**), du médecin (**medecin_id**), le titre (**title**), la date (**date**) et la description (**description**).
- Table **patient** : Table des patients enregistrés, liés à un utilisateur via **user_id** et enregistrant des détails tels que l'adresse (**adlibelle**, **adcp**, **adcity**, **adcountry**).
- Table **prescription** : Enregistre les prescriptions médicales émises par les médecins, liant un médecin (**medecin_id**) à un patient (**patient_id**) pour une période définie par une date de début (**start_date**) et une date de fin (**end_date**).
- Table **speciality** : Liste des spécialités médicales disponibles avec des champs pour le nom (**name**) et une description optionnelle (**description**).
- Table **stay** : Enregistre les séjours des patients pour des spécialités spécifiques (**speciality_id**) avec des médecins (**medecin_id**), incluant des détails comme la raison (**reason**), la description (**description**), les dates de début (**start_date**) et de fin (**end_date**), et une validation (**validate**).

- Table **user** : Table principale des utilisateurs du système, enregistrant des informations telles que l'email (**email**), les rôles (**roles**), le mot de passe (**password**), le prénom (**firstname**) et le nom (**lastname**).

Contraintes et clés étrangères

Les clés primaires et les index sont définis pour assurer l'unicité et l'efficacité des recherches.

Des contraintes de clés étrangères sont utilisées pour maintenir l'intégrité référentielle entre les tables, garantissant ainsi la cohérence des données.

Cette structure de base de données, associée aux contraintes et aux clés étrangères, fournit un cadre solide pour la gestion efficace des données de l'application "SoigneMoi".