# 

# **Table des Matières**

# **Introduction**

# **Analyse des besoins**

# **Conception de l'application MedManager**

# **Développement et Implémentation**

# **Conclusion**

# **Annexes**

# 

# 

# 

# 

# 

# 

# 

# **Introduction**

Le secteur pharmaceutique occupe une place prépondérante dans le domaine de la santé mondiale, contribuant significativement au développement de nouveaux traitements et à l'amélioration de la qualité de vie des patients. Au cours des dernières décennies, ce secteur a été marqué par de nombreuses fusions et acquisitions, donnant naissance à des entreprises de grande envergure. Ces regroupements visent à mutualiser les ressources, accélérer la recherche et renforcer la présence sur le marché international.

Galaxy Swiss Bourdin (GSB) est le fruit de la fusion en 2009 entre le géant américain Galaxy, spécialisé dans les maladies virales telles que le SIDA et les hépatites, et le conglomérat européen Swiss Bourdin, reconnu pour ses médicaments conventionnels. Cette union stratégique a permis à GSB de s'imposer comme un leader du secteur pharmaceutique, avec un siège administratif établi à Paris et un siège social basé à Philadelphie, aux États-Unis.

Dans ce contexte, la gestion efficace et sécurisée des données médicales est devenue un enjeu majeur. Les médecins ont besoin d'outils fiables pour gérer les informations des patients, prescrire des ordonnances en tenant compte des antécédents médicaux, des allergies, et des interactions médicamenteuses potentielles. C'est dans cette optique que l'application **MedManager** a été développée.

**MedManager** est une application de gestion des ordonnances médicales conçue pour répondre aux besoins spécifiques des médecins de GSB. Elle vise à fournir une plateforme sécurisée pour l'authentification des utilisateurs, la gestion des profils patients, la création et le suivi des ordonnances, ainsi que l'accès à une base de données exhaustive de médicaments. L'application met l'accent sur la facilité d'utilisation, la sécurité des données et la précision des informations médicales, tout en respectant les réglementations en vigueur telles que le RGPD.

Ce rapport présente le processus complet de développement de l'application **MedManager**, depuis l'analyse des besoins jusqu'aux phases de conception, d'implémentation et de tests. Il met en lumière les choix technologiques effectués, notamment l'utilisation du framework ASP.NET MVC, ainsi que les outils et bibliothèques qui ont permis de concrétiser ce projet. Enfin, il aborde les défis rencontrés, les solutions apportées et les perspectives d'évolution de l'application.

**Analyse des besoins**

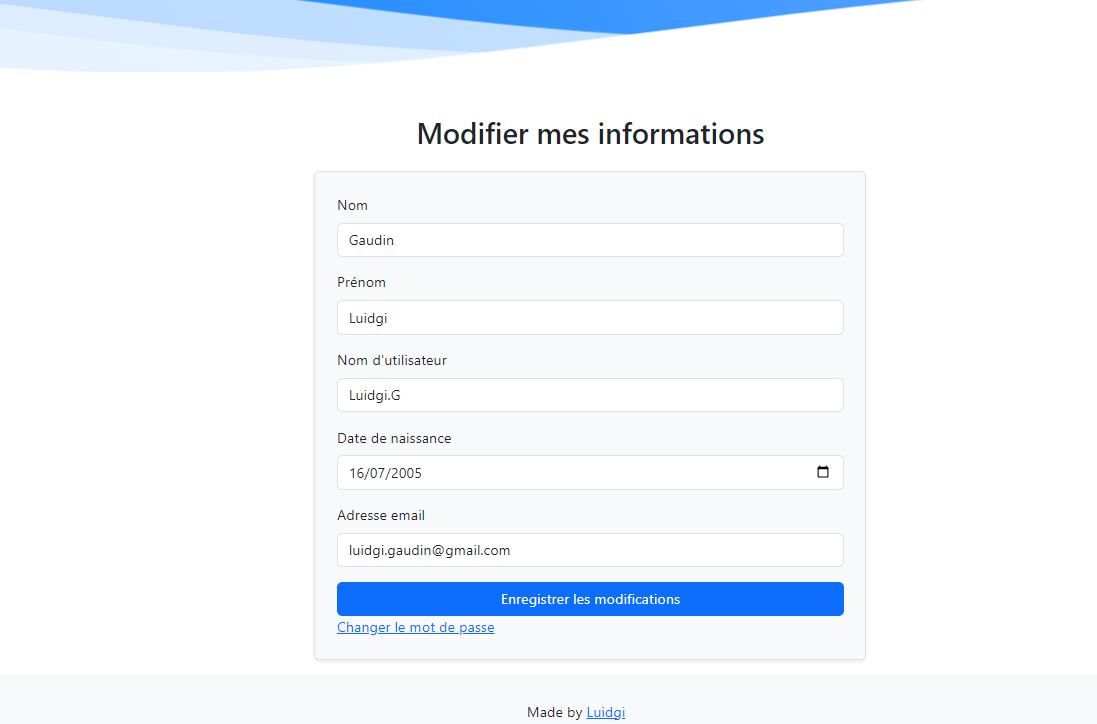
## **Identification des utilisateurs**

L'application **MedManager** est destinée aux médecins du laboratoire Galaxy Swiss Bourdin (GSB). Ces professionnels de santé ont besoin d'un outil performant pour gérer efficacement les informations médicales de leurs patients, tout en assurant la confidentialité et la sécurité des données sensibles.

## **Besoins fonctionnels**

### **1. Authentification et gestion des utilisateurs**

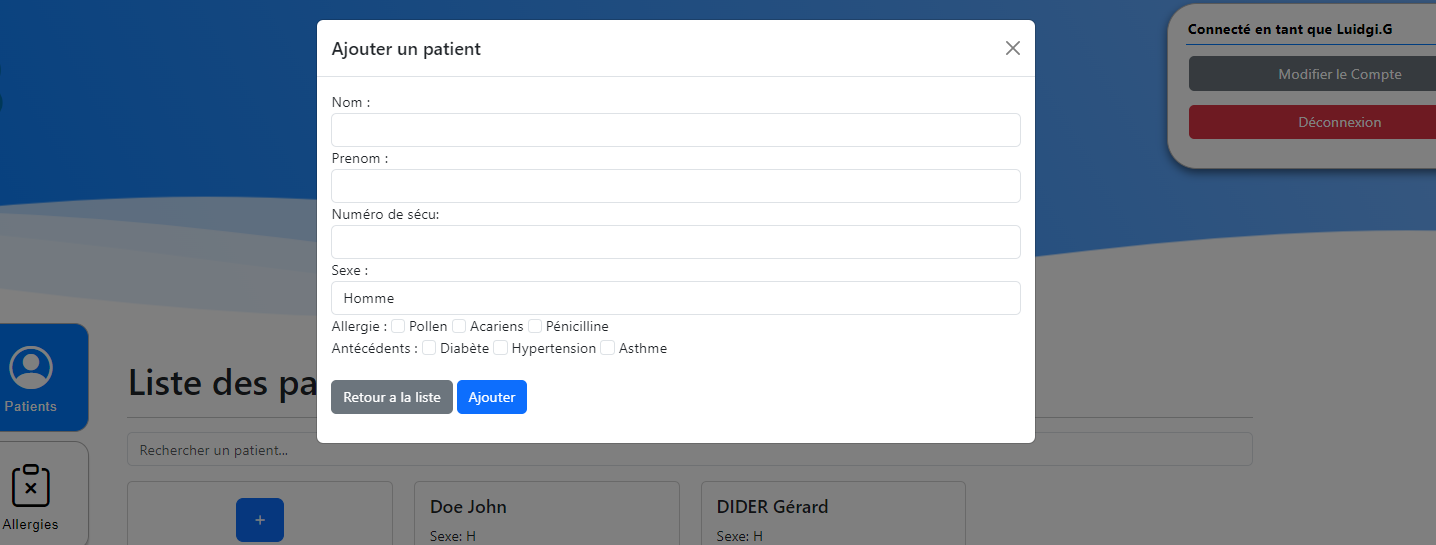
* **Sécurité de l'accès** :
  + Mise en place d'un système d'authentification sécurisé pour garantir que seuls les médecins autorisés peuvent accéder à l'application.
  + Utilisation de **Identity** (paquet NuGet) pour gérer l'authentification et l'autorisation des utilisateurs.
* **Gestion du profil médecin** :
  + Les médecins doivent pouvoir consulter et modifier leurs informations personnelles.

**

### 

### **2. Gestion des patients**

* **Liste des patients avec fonction de recherche** :
  + Affichage d'une liste complète des patients avec possibilité de rechercher par nom, prénom ou autre critère pertinent.
* **Fiche détaillée du patient** :
  + Accès aux informations détaillées du patient, y compris les antécédents médicaux et les allergies.
* **Formulaire d'ajout/modification de patient** :
  + Possibilité d'ajouter de nouveaux patients ou de modifier les informations existantes.

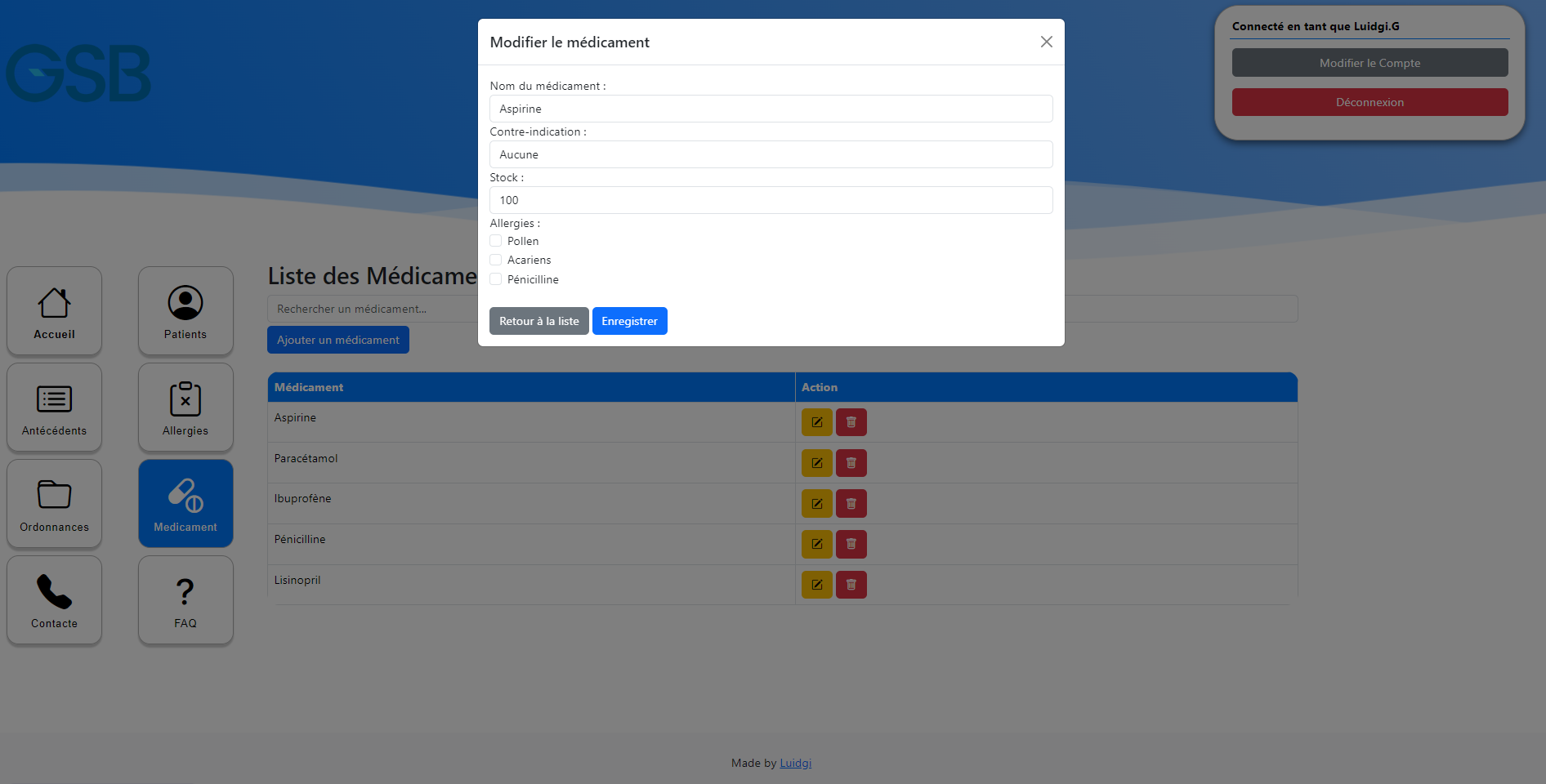
**

### **3. Gestion des ordonnances**

* **Création d'une nouvelle ordonnance** :
  + Interface intuitive pour rédiger une nouvelle ordonnance, incluant la sélection des médicaments et la saisie des posologies.
* **Modification d'une ordonnance existante** :
  + Possibilité de mettre à jour ou d'annuler une ordonnance précédemment créée.
* **Visualisation détaillée d'une ordonnance** :
  + Affichage complet des informations de l'ordonnance, y compris les instructions spéciales.
* **Exportation en PDF** :
  + Utilisation de **iText7** (paquet NuGet) pour permettre l'exportation des ordonnances au format PDF.

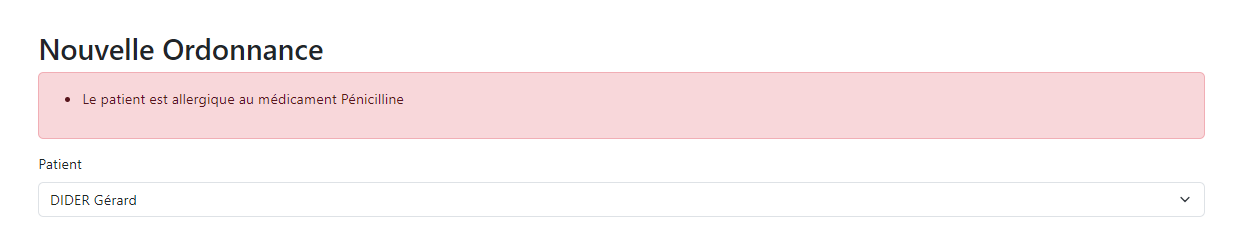
### **4. Base de données des médicaments**

* **Liste des médicaments avec fonction de recherche** :
  + Accès à une base de données exhaustive des médicaments disponibles.
* **Informations sur les contre-indications et interactions** :
  + Alertes sur les interactions potentielles avec d'autres médicaments ou allergies du patient.

*.*

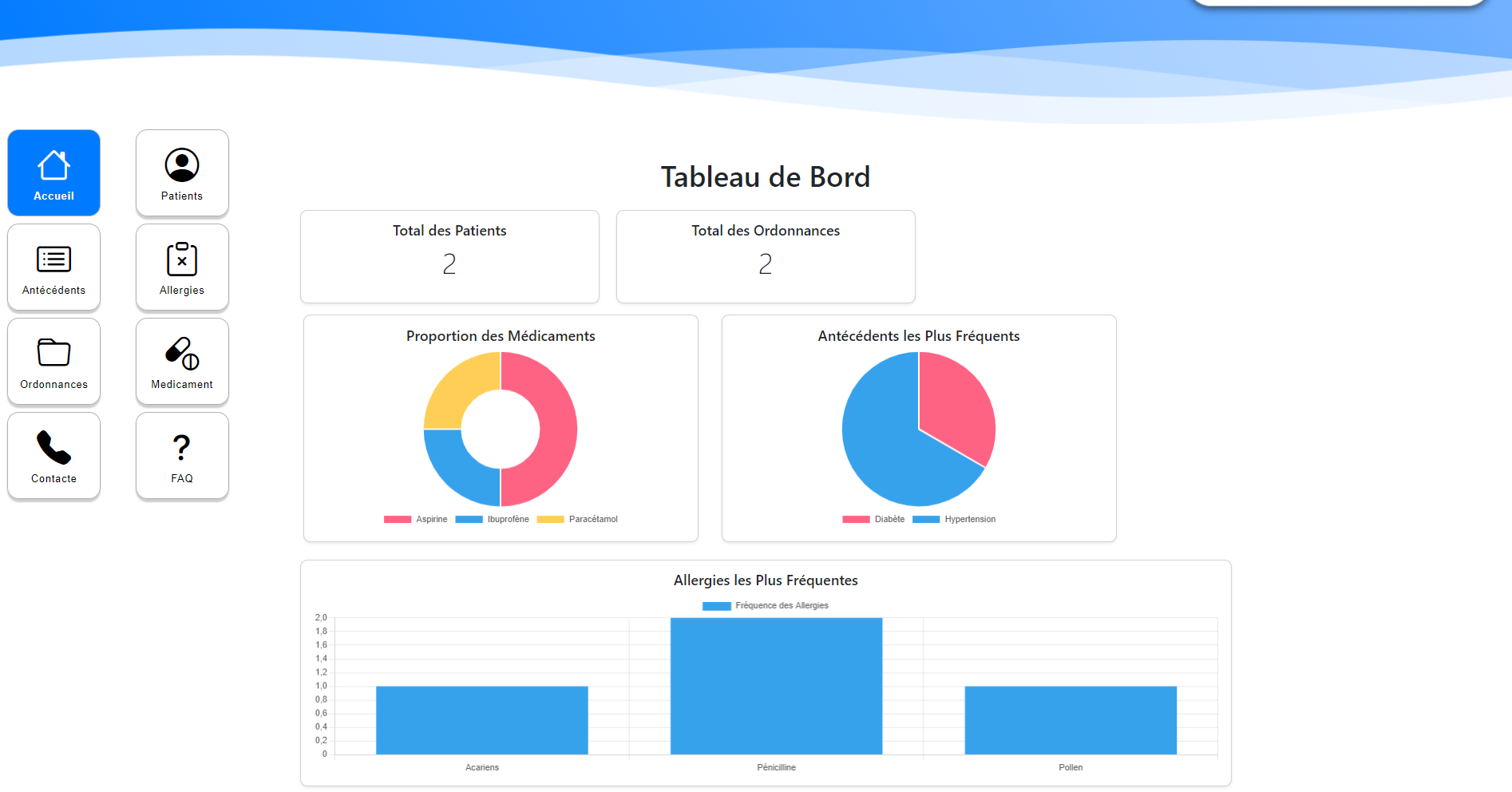
### **5. Création d'ordonnance assistée**

* **Interface de sélection des médicaments** :
  + Sélection facile des médicaments à partir de la base de données.
* **Saisie de la posologie et de la durée du traitement** :
  + Champs dédiés pour indiquer la posologie exacte et la durée prévue.
* **Ajout d'instructions spéciales** :
  + Possibilité d'ajouter des notes ou des instructions particulières pour le patient.
* **Système d'alerte pour les interactions** :
  + Notifications automatiques en cas de détection d'interactions médicamenteuses ou de contre-indications.

**

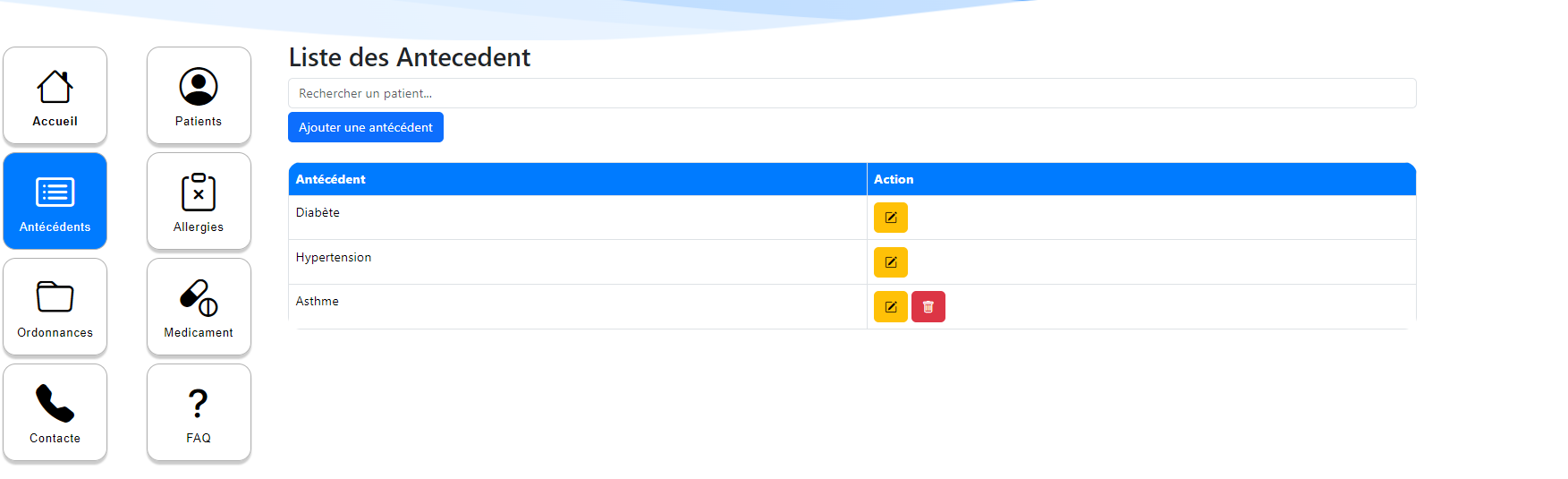
### **6. Tableau de bord du médecin**.

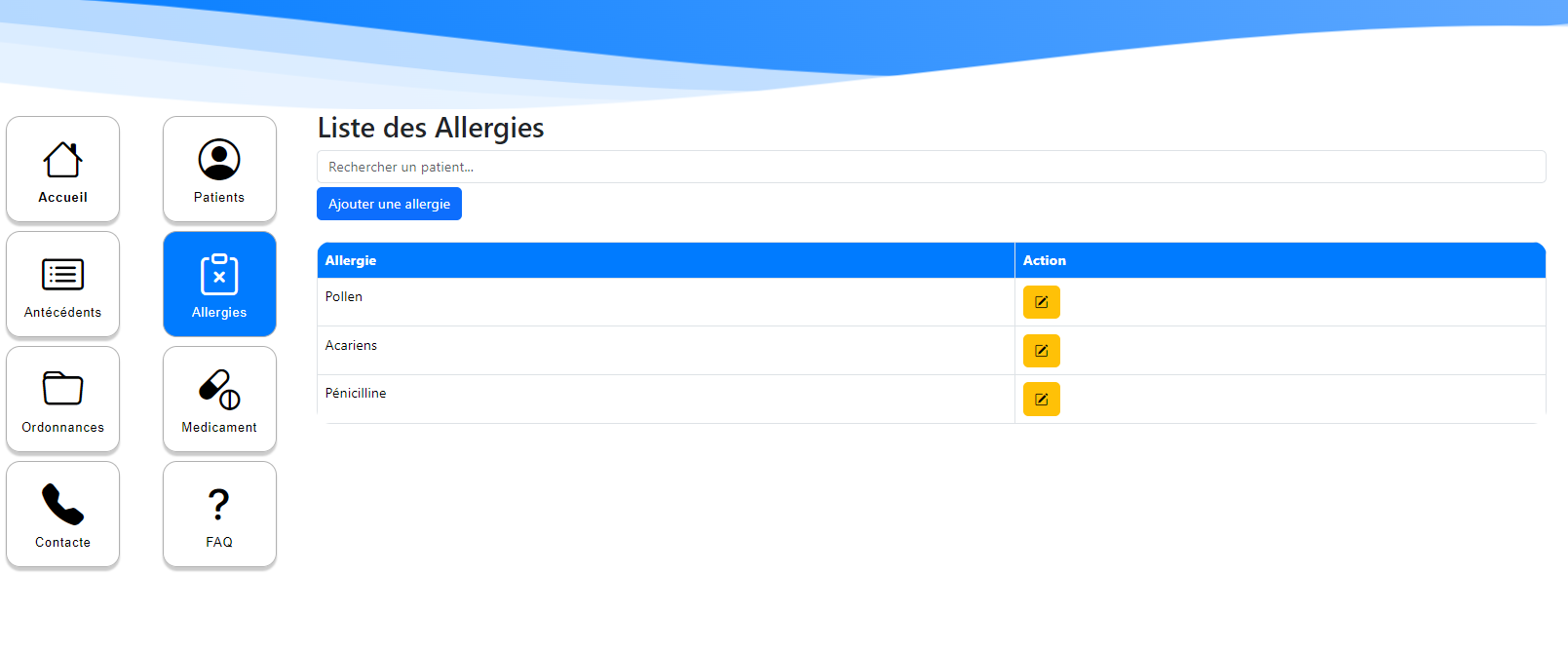
* **Statistique de médicaments et antécédents** :
  + Suivi de la quantité de médicaments et d’antécédents .

**

### **7. Gestion des allergies et antécédents**

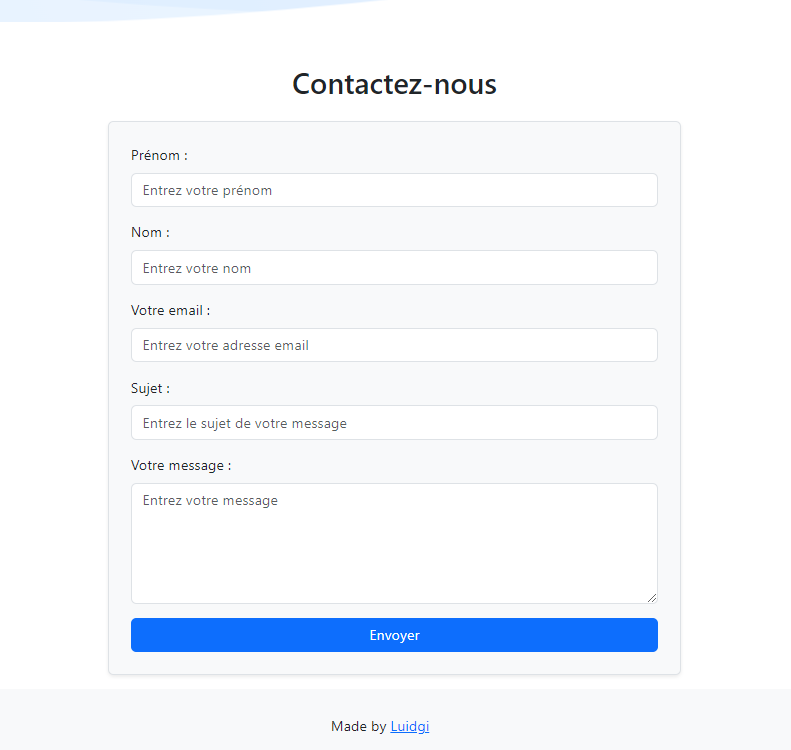
* **Interface d'ajout/modification des allergies** :
  + Mise à jour facile des allergies connues du patient.
* **Interface d'ajout/modification des antécédents médicaux** :
  + Enregistrement des antécédents pour une prise en charge globale du patient.

**

**

**8. Aide et support**

* **Page de FAQ** :
  + Réponses aux questions fréquentes.
* **Formulaire de contact pour le support technique** :
  + Moyen de communication direct avec l'équipe de support pour signaler des problèmes ou des suggestions.

**

## **Besoins non fonctionnels**

* **Sécurité des données et conformité réglementaire** :
  + Respect des normes telles que le RGPD pour la protection des données personnelles.
  + Chiffrement des données sensibles et utilisation de protocoles sécurisés.
* **Performance et fiabilité** :
  + Temps de réponse rapide et disponibilité élevée de l'application.
* **Facilité d'utilisation et ergonomie** :
  + Interface utilisateur intuitive avec une navigation simplifiée.
* **Scalabilité** :
  + Capacité à gérer une augmentation du nombre d'utilisateurs et de données sans dégradation des performances.
* **Maintenabilité** :
  + Code structuré pour faciliter les mises à jour futures.

## **Contraintes**

* **Technologiques** :
  + Utilisation du framework **ASP.NET MVC** pour le développement de l'application.
  + Intégration avec **Entity Framework** pour la gestion de la base de données relationnelle.
  + Utilisation de paquets NuGet tels que **Identity** pour l'authentification et **iText7** pour la génération de PDF.

# 

# 

# 

# 

# 

# 

# **Conception de l'application MedManager**

## **Choix technologiques**

Pour répondre aux besoins identifiés et aux contraintes du projet, plusieurs choix technologiques ont été effectués :

* **ASP.NET MVC** : Ce framework de Microsoft permet de créer des applications web robustes en séparant clairement les responsabilités entre le Modèle, la Vue et le Contrôleur. Il facilite le développement, la maintenance et les tests de l'application.
* **C#** : Langage de programmation orienté objet utilisé pour développer avec ASP.NET MVC, offrant une syntaxe claire et une intégration étroite avec le framework .NET.
* **Entity Framework** : Cet outil d'ORM (Object-Relational Mapping) simplifie l'interaction avec la base de données en permettant de manipuler des objets plutôt que des requêtes SQL. Il facilite les opérations CRUD (Create, Read, Update, Delete) et assure une correspondance entre le modèle objet et le schéma relationnel de la base de données.
* **Identity** : Ce paquet NuGet gère l'authentification et l'autorisation des utilisateurs dans les applications ASP.NET. Il offre des fonctionnalités prêtes à l'emploi pour la gestion des utilisateurs, des rôles, du stockage sécurisé des mots de passe, et bien plus encore.
* **iText7** : Bibliothèque puissante pour la génération de documents PDF. Elle permet de créer des ordonnances au format PDF de manière dynamique, en respectant les normes professionnelles de présentation.
* **SQL Server** : Système de gestion de base de données relationnelle choisi pour stocker de manière sécurisée les données de l'application, telles que les informations des patients, des médecins, des médicaments et des ordonnances.

## **Architecture de l'application**

L'application **MedManager** est conçue selon le modèle architectural MVC (Modèle-Vue-Contrôleur) qui sépare l'application en trois composants principaux :

1. **Modèle** : Représente les données de l'application et la logique métier. Il contient les classes qui modélisent les entités telles que les patients, les médecins, les médicaments et les ordonnances.
2. **Vue** : Gère l'affichage des données à l'utilisateur. Les vues sont responsables de la présentation des informations sous forme d'interfaces utilisateur intuitives.
3. **Contrôleur** : Agit comme un intermédiaire entre le Modèle et la Vue. Il reçoit les entrées de l'utilisateur via la Vue, traite les demandes, interagit avec le Modèle, puis retourne une réponse appropriée à la Vue.

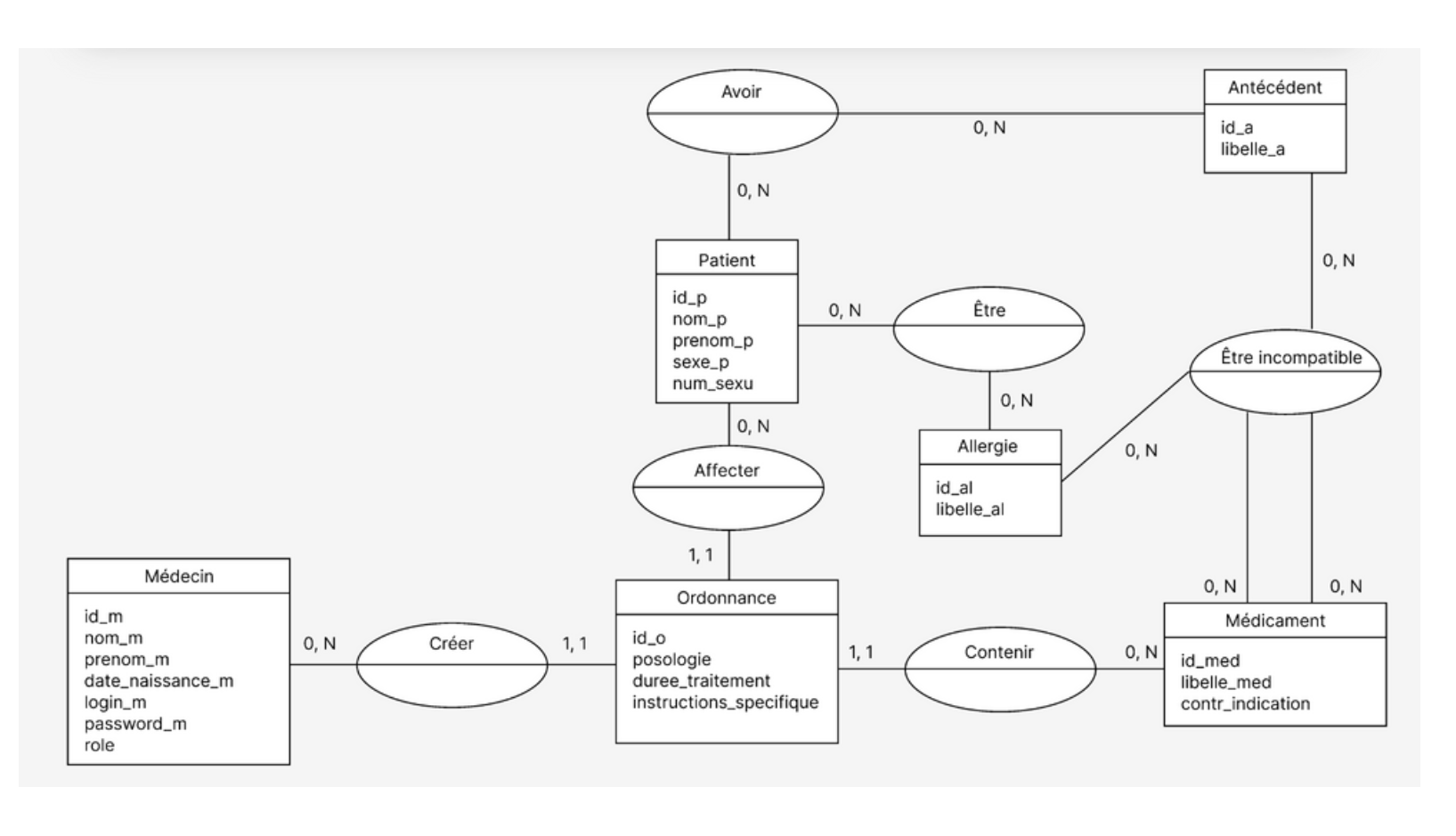
## **Modélisation des données**

### **Modèle Conceptuel de Données (MCD)**

Le MCD est utilisé pour représenter de manière conceptuelle les entités et les relations entre elles au sein de l'application. Les principales entités identifiées sont :

* **Médecin** : Représente les utilisateurs de l'application. Chaque médecin possède des informations telles que l'identifiant, le nom, le prénom, l'adresse e-mail, et les informations d'authentification.
* **Patient** : Contient les informations personnelles du patient, ses antécédents médicaux et ses allergies.
* **Médicament** : Inclut les détails sur chaque médicament, y compris le nom, la description, les contre-indications et les interactions potentielles.
* **Ordonnance** : Correspond à une prescription faite par un médecin pour un patient. Elle inclut les médicaments prescrits, la posologie, la durée du traitement et les instructions spéciales.
* **Allergie** : Liste des allergies connues qui peuvent être associées aux patients.
* **Antécédent Médical** : Historique médical du patient, y compris les maladies antérieures et les conditions chroniques.

*Diagramme du Modèle Conceptuel de Données (MCD).*

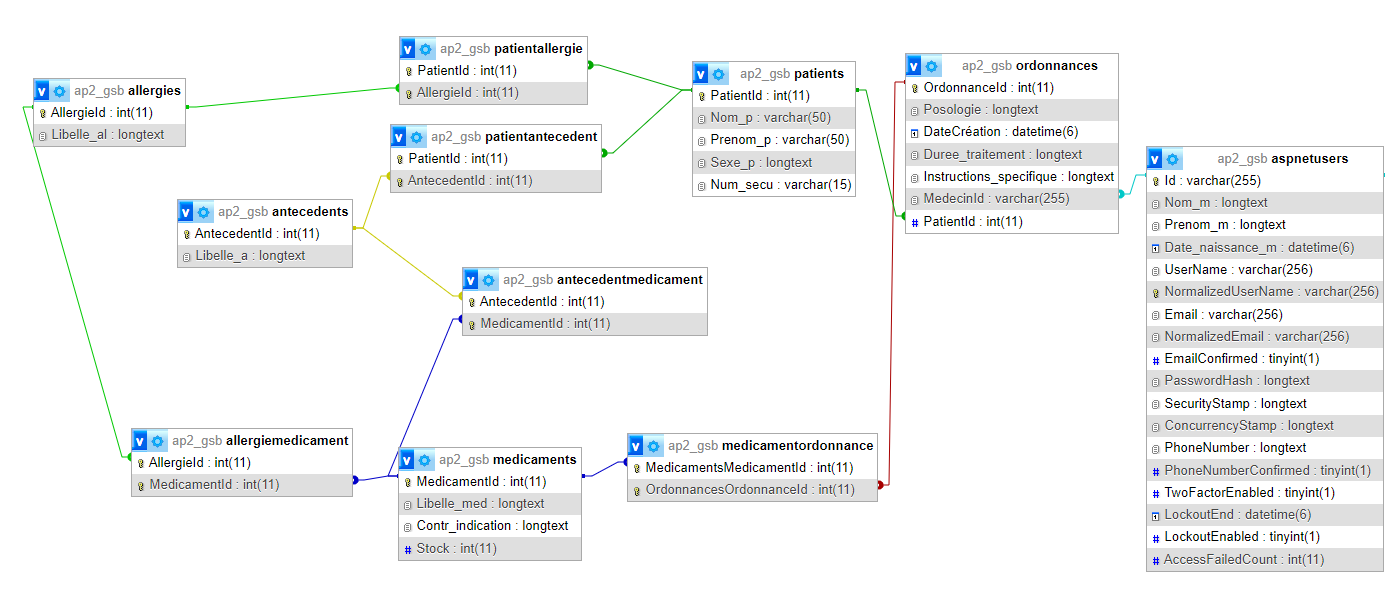
**

### **Modèle Logique de Données (MLD)**

Le MLD traduit le MCD en un modèle plus proche de l'implémentation physique dans la base de données relationnelle. Il définit les tables, les colonnes, les types de données, les clés primaires et étrangères.

* **Tables principales** :
  + **Medecins** (MedecinID, Nom, Prenom, Email, MotDePasse, etc.)
  + **Patients** (PatientID, Nom, Prenom, DateNaissance, Sexe, etc.)
  + **Medicaments** (MedicamentID, Nom, Description, ContreIndications, etc.)
  + **Ordonnances** (OrdonnanceID, MedecinID, PatientID, DateCreation, etc.)
  + **OrdonnanceMedicaments** (OrdonnanceID, MedicamentID, Posologie, Duree, InstructionsSpeciales)
  + **Allergies** (AllergieID, Description)
  + **PatientAllergies** (PatientID, AllergieID)
  + **AntecedentsMedicaux** (AntecedentID, Description)
  + **PatientAntecedents** (PatientID, AntecedentID)

*Diagramme du Modèle Logique de Données (MLD).*

**

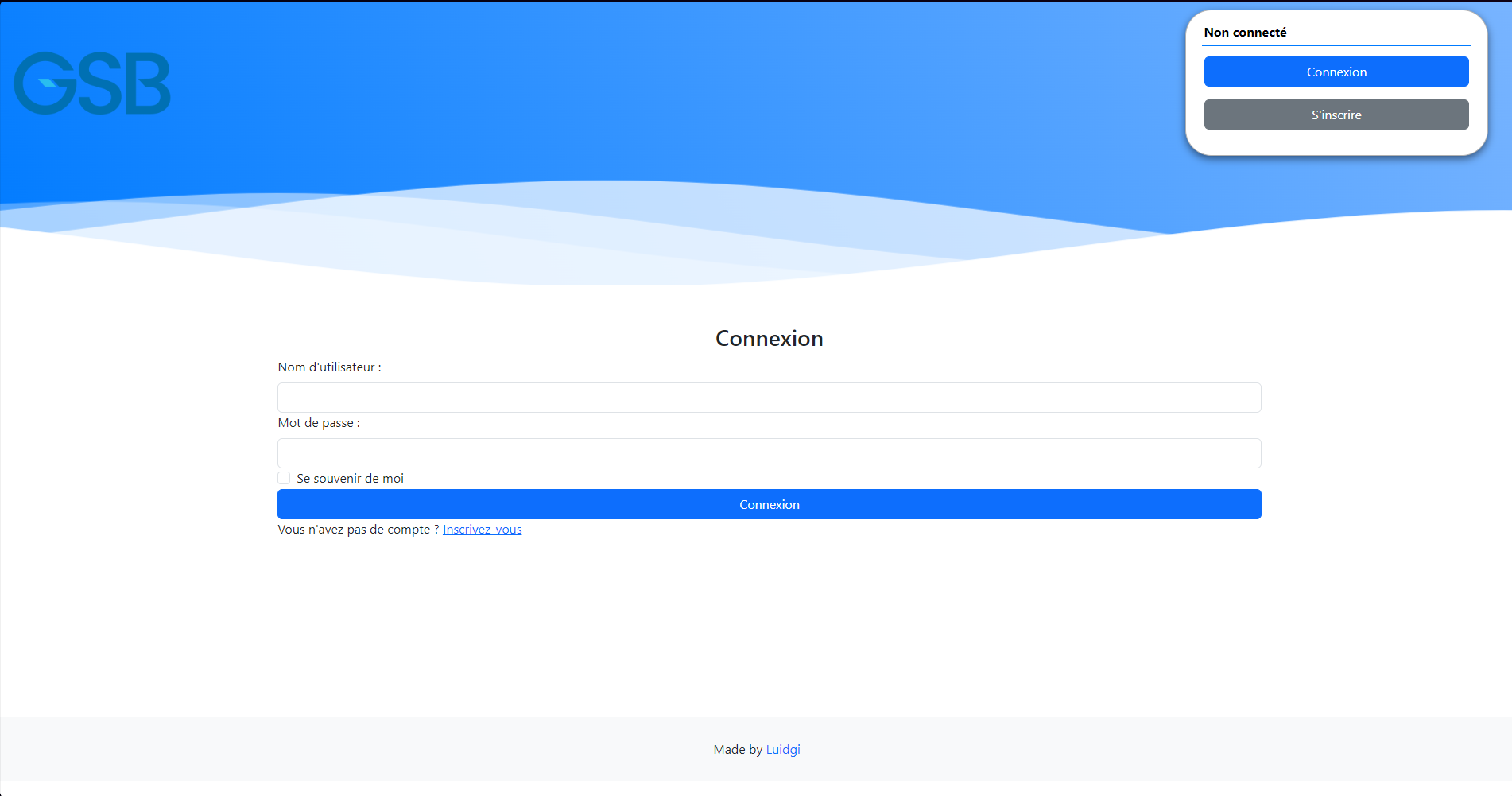
## **Conception des interfaces utilisateur**

La conception des interfaces se focalise sur l'ergonomie et la facilité d'utilisation pour les médecins. Les principes suivants ont été appliqués :

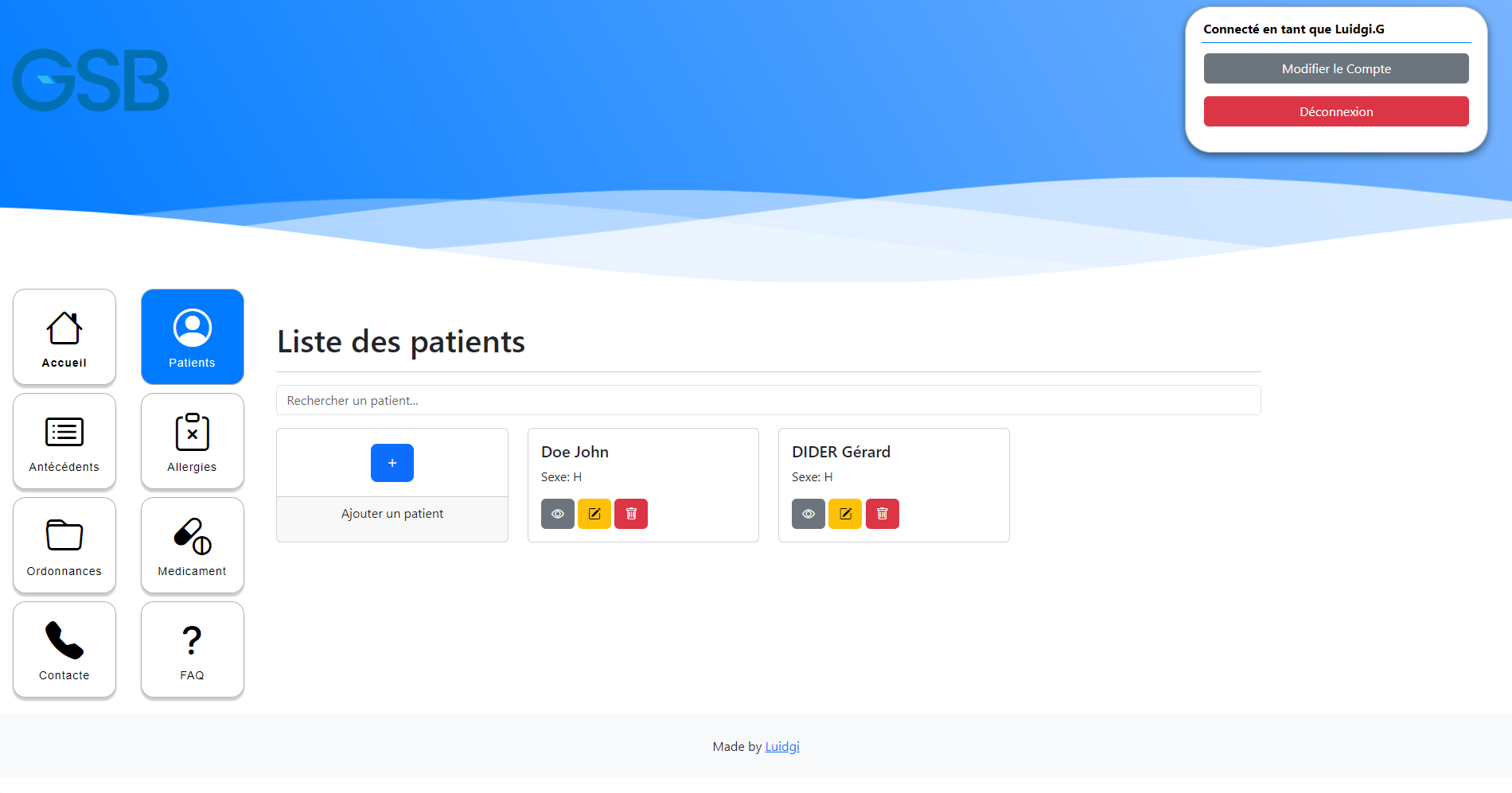
* **Simplicité** : Interfaces épurées avec les informations essentielles mises en avant.
* **Cohérence** : Uniformité dans le design et la navigation à travers les différentes pages.
* **Accessibilité** : Utilisation de polices lisibles, de contrastes adéquats et de dispositions adaptatives pour différents dispositifs.

### **Maquettes des principales pages**

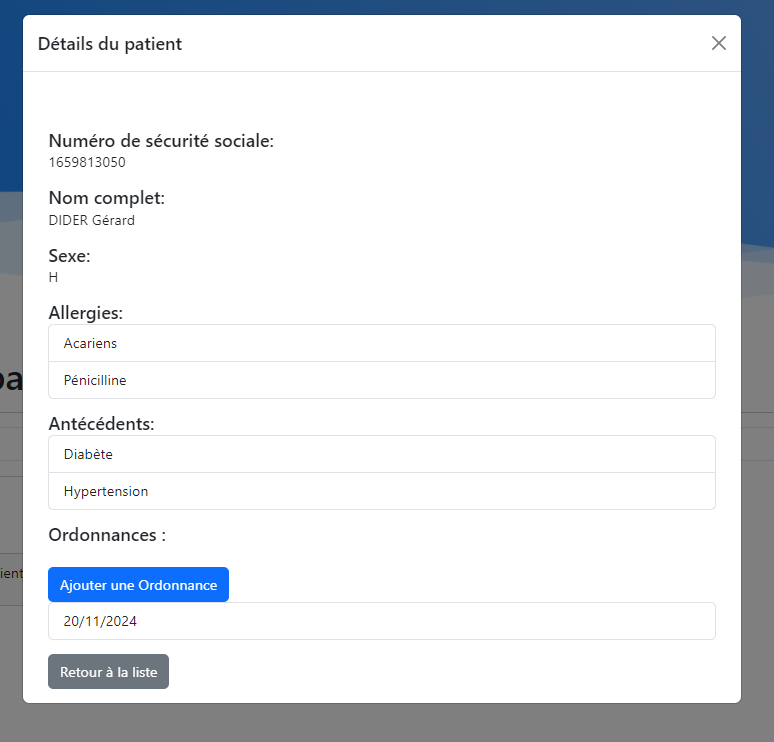
1. **Écran de connexion**
   * Champ pour l'adresse e-mail et le mot de passe.

**

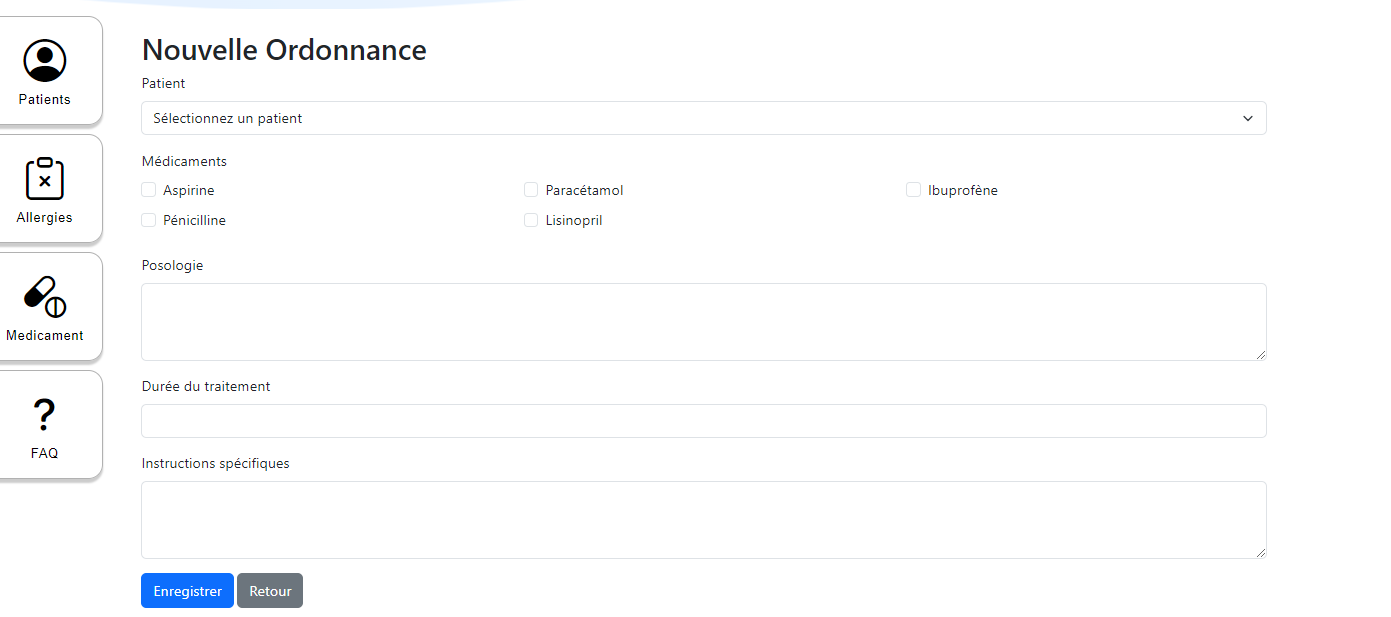
1. **Gestion des patients**
   * Liste des patients avec fonction de recherche.
   * Bouton pour ajouter un nouveau patient.
   * Accès aux fiches détaillées des patients.

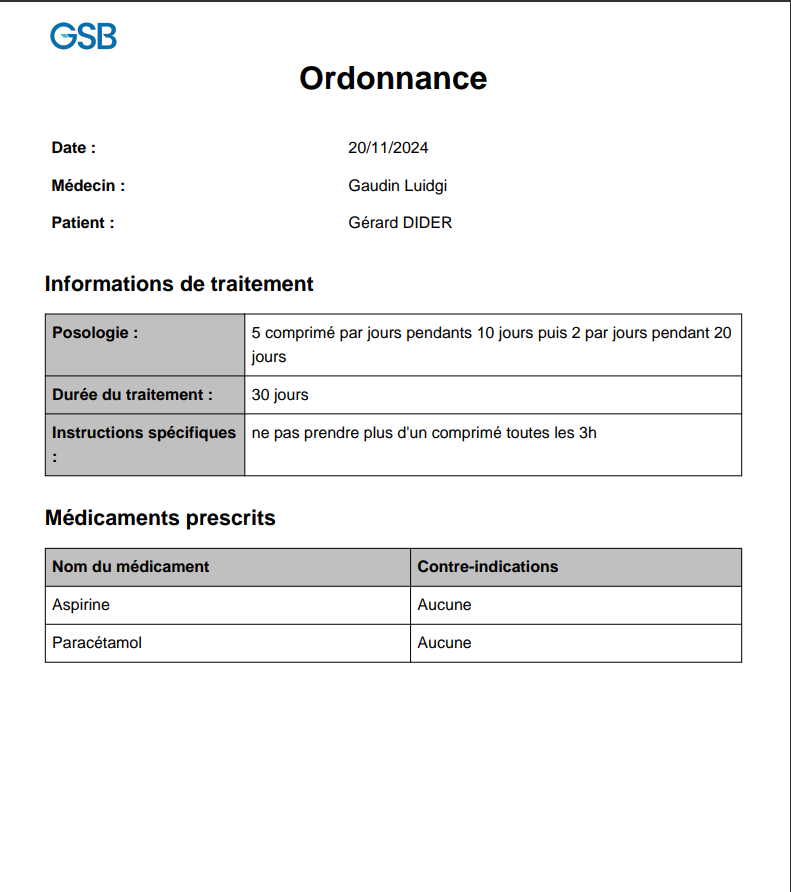
**

1. **Fiche détaillée du patient**
   * Informations personnelles du patient.
   * Sections pour les antécédents médicaux et les allergies.
   * Bouton pour créer une nouvelle ordonnance.

**

1. **Création d'une ordonnance**
   * Sélection du patient et des médicaments.
   * Saisie de la posologie, de la durée du traitement et des instructions spéciales.
   * Alertes automatiques en cas d'interactions médicamenteuses ou de contre-indications.

**

1. **Exportation en PDF**
   * Aperçu de l'ordonnance.
   * Bouton pour exporter et télécharger le document au format PDF.

## 

## 

## 

## 

## 

## 

## 

## 

## **Explication des paquets NuGet utilisés**

### **Identity**

* **Fonctionnalité** : Gestion de l'authentification et de l'autorisation des utilisateurs.
* **Utilisation dans MedManager** :
  + Stockage sécurisé des informations d'authentification des médecins.
  + Gestion des rôles et des permissions pour contrôler l'accès aux différentes parties de l'application.
  + Implémentation des fonctionnalités de connexion, déconnexion et récupération de mot de passe.

### **Entity Framework**

* **Fonctionnalité** : Outil d'ORM pour interagir avec la base de données de manière objet.
* **Utilisation dans MedManager** :
  + Mapping des classes du modèle aux tables de la base de données.
  + Exécution des opérations CRUD sans écrire de requêtes SQL manuelles.
  + Gestion des migrations pour maintenir la synchronisation entre le modèle et la base de données.

### **iText7**

* **Fonctionnalité** : Bibliothèque pour la création et la manipulation de documents PDF.
* **Utilisation dans MedManager** :
  + Génération des ordonnances au format PDF.
  + Personnalisation de la mise en page, incluant le logo du laboratoire, les informations du médecin et du patient, les détails de la prescription et les instructions spéciales.

## **Sécurité et conformité**

La sécurité des données médicales est une priorité absolue. Plusieurs mesures ont été mises en place :

* **Chiffrement des mots de passe** : Utilisation de techniques de hachage sécurisées pour stocker les mots de passe des utilisateurs.
* **Protocoles sécurisés** : Utilisation du protocole HTTPS pour chiffrer les communications entre le client et le serveur.
* **Contrôle des accès** : Mise en place de politiques d'autorisation pour restreindre l'accès aux informations sensibles.

# **Développement et Implémentation**

## **Gestion de l'authentification et des utilisateurs**

### **Mise en place de la sécurité avec Identity**

Pour garantir une authentification sécurisée et une gestion efficace des utilisateurs, nous avons intégré le paquet NuGet **ASP.NET Identity**. Ce framework offre une solution complète pour la gestion des comptes utilisateurs, incluant l'inscription, la connexion, la gestion des mots de passe et la récupération en cas d'oubli.

**Configuration initiale :**

* **Installation du paquet :** Le paquet **Microsoft.AspNet.Identity.EntityFramework** a été ajouté au projet via le gestionnaire de packages NuGet.
* **Configuration du contexte de données :** Un contexte de données personnalisé, dérivé de IdentityDbContext, a été créé pour intégrer les tables d'Identity dans notre base de données existante.
* **Modèle utilisateur personnalisé :** Une classe ApplicationUser dérivant de IdentityUser a été créée pour ajouter des propriétés spécifiques aux médecins, telles que le nom, le prénom et les informations de contact.

**Gestion des mots de passe :**

* **Hashage sécurisé :** Les mots de passe sont hachés en utilisant des algorithmes sécurisés (par défaut, le HMAC-SHA256 avec un sel unique).
* **Politiques de mot de passe :** Des règles de complexité ont été définies pour renforcer la sécurité, exigeant un minimum de caractères, l'inclusion de chiffres et de caractères spéciaux.

### **Gestion des sessions utilisateur**

* **Authentification par cookie :** L'application utilise des cookies d'authentification pour maintenir la session de l'utilisateur.
* **Protection contre les attaques CSRF :** Des tokens anti-CSRF sont utilisés pour protéger les formulaires contre les soumissions non autorisées.

## **Fonctionnalités principales**

### **Implémentation de la gestion des patients**

**Utilisation d'Entity Framework pour les opérations CRUD**

* **Modèles de données :** Les classes Patient, Allergie et AntecedentMedical ont été définies pour représenter les entités du domaine.
* **Migrations :** Entity Framework Code First Migrations a été utilisé pour créer et mettre à jour la base de données en fonction des modèles.
* **Contrôleurs et vues :** Des contrôleurs MVC ont été créés pour gérer les actions CRUD, avec des vues correspondantes pour l'interface utilisateur.

**Interface utilisateur**

* **Liste des patients :** Affichage des patients avec une barre de recherche pour filtrer par nom, prénom.
* **Fiche détaillée du patient :** Présentation des informations complètes du patient, incluant les antécédents médicaux et les allergies, avec des options pour modifier ou ajouter des informations.

### **Création, modification et visualisation des ordonnances**

**Modélisation des ordonnances**

* **Entité Ordonnance :** La classe Ordonnance contient des propriétés telles que l'identifiant du médecin, du patient, la date de création, et une collection de médicaments prescrits.
* **Relation avec les médicaments :** Une table de jonction OrdonnanceMedicament gère la relation plusieurs-à-plusieurs entre les ordonnances et les médicaments, en incluant la posologie, la durée et les instructions spéciales.

**Génération et exportation des ordonnances en PDF avec iText7**

* **Intégration de iText7 :** Le paquet **iText7** a été ajouté pour permettre la création dynamique de documents PDF.
* **Création du document PDF :** Lorsqu'une ordonnance est validée, un PDF est généré avec le logo de GSB, les informations du médecin et du patient, la liste des médicaments avec leurs posologies, et les instructions spéciales.
* **Téléchargement et impression :** Le médecin peut télécharger le PDF pour l'imprimer ou l'envoyer électroniquement au patient.

## **Intégration de la base de données des médicaments**

### **Sources des données pharmaceutiques**

* **Base de données interne :** Une base de données a été créée pour stocker les informations sur les médicaments, incluant les contre-indications.
* **Mise à jour régulière :** Des mécanismes ont été prévus pour mettre à jour les données pharmaceutiques à partir de sources fiables.

### **Utilisation d'Entity Framework pour accéder aux données**

* **Modèle Médicament :** La classe Medicament représente les médicaments, avec des propriétés pour le nom, la description, les contre-indications, etc.
* **Requêtes LINQ :** Les opérations de recherche et de filtrage utilisent LINQ pour interroger efficacement la base de données.

### **Recherche et affichage des informations sur les médicaments**

* **Interface utilisateur :** Une page permet de rechercher des médicaments par nom.
* **Fiche détaillée :** Présentation des informations complètes du médicament, incluant les contre-indications et les interactions médicamenteuses.

## **Gestion des alertes et notifications**

### **Détection des interactions médicamenteuses et contre-indications**

* **Vérification automatique :** Lors de la prescription, l'application vérifie automatiquement les interactions potentielles entre les médicaments sélectionnés et les allergies ou antécédents du patient.
* **Algorithme de détection :** Un algorithme compare les médicaments prescrits avec les données des contre-indications et des interactions stockées dans la base de données.

### **Système d'avertissement pour le médecin lors de la prescription**

* **Alertes en temps réel :** Si une interaction ou une contre-indication est détectée, une alerte s'affiche immédiatement à l'écran empêchant le médecin de prescrire le médicament.

## **Tableau de bord du médecin**

### **Mise en place des vues d'ensemble**

* **Statistiques :** Des graphiques faisant des ratio d’utilisation de Médicaments ou d’antécédents recensé avec le nombre de patients.

## **Aide et support intégrés**

### **Implémentation de la page de FAQ et du formulaire de contact**

* **FAQ :** Réponses aux questions fréquemment posées, classées par catégories.
* **Support technique :** Un formulaire permet aux médecins de soumettre des demandes d'assistance par mail grâce à **FORMSPREE**.

## **Paramètres de l'application**

### **Configuration personnalisée des alertes**

* **Personnalisation :** Les médecins peuvent ajuster les paramètres des alertes selon leurs préférences, comme le niveau de sensibilité aux interactions médicamenteuses.
* **Sauvegarde des préférences :** Les réglages sont enregistrés pour chaque utilisateur et appliqués lors des sessions suivantes.

### **Gestion des droits d'accès**

* **Contrôle d'accès :** Bien que l'application soit principalement utilisée par des médecins, la gestion des droits permet d'ajouter d'autres rôles si nécessaire.
* **Sécurité renforcée :** Les fonctionnalités sensibles sont protégées par des autorisations spécifiques pour prévenir tout accès non autorisé.

# **Conclusion**

Le projet **MedManager** a été conçu et développé pour répondre aux besoins spécifiques des médecins du laboratoire Galaxy Swiss Bourdin (GSB) en matière de gestion des ordonnances médicales. Grâce à une analyse approfondie des besoins fonctionnels et non fonctionnels, nous avons pu élaborer une application robuste, sécurisée et intuitive, intégrant les technologies modernes du framework ASP.NET MVC.

Les objectifs principaux du projet ont été atteints :

* **Sécurité et confidentialité des données** : En utilisant **Identity** pour l'authentification et l'autorisation, ainsi que des protocoles de sécurité renforcés, nous avons assuré la protection des informations sensibles des patients et des médecins.
* **Gestion efficace des patients et des ordonnances** : L'application permet une gestion complète des dossiers patients, des antécédents médicaux, des allergies, et offre un système intuitif pour la création et le suivi des ordonnances.
* **Accès à une base de données exhaustive des médicaments** : L'intégration d'une base de données des médicaments avec des informations détaillées sur les contre-indications et les interactions médicamenteuses aide les médecins à prendre des décisions éclairées.
* **Amélioration de l'expérience utilisateur** : L'interface utilisateur a été conçue pour être ergonomique et facile à naviguer, avec des fonctionnalités telles que le tableau de bord personnalisé, les alertes en temps réel et l'aide intégrée.

**Compétences acquises** :

* Maîtrise du framework ASP.NET MVC et des technologies associées.
* Intégration de bibliothèques externes telles que **Entity Framework** et **iText7**.
* Application des principes de sécurité informatique et de protection des données.
* Gestion de projet logiciel, de l'analyse des besoins à la mise en production.

**Difficultés rencontrées** :

* **Gestion des interactions médicamenteuses** : La complexité des interactions possibles entre les médicaments a nécessité le développement d'un algorithme sophistiqué et la mise à jour régulière des données pharmaceutiques.
* **Conformité réglementaire** : Assurer la conformité avec les réglementations telles que le RGPD a exigé une attention particulière aux détails et des consultations avec des experts juridiques.
* **Optimisation des performances** : L'application devait rester réactive malgré la quantité de données traitées, ce qui a nécessité des optimisations au niveau de la base de données et du code.

**Perspectives d'amélioration** :

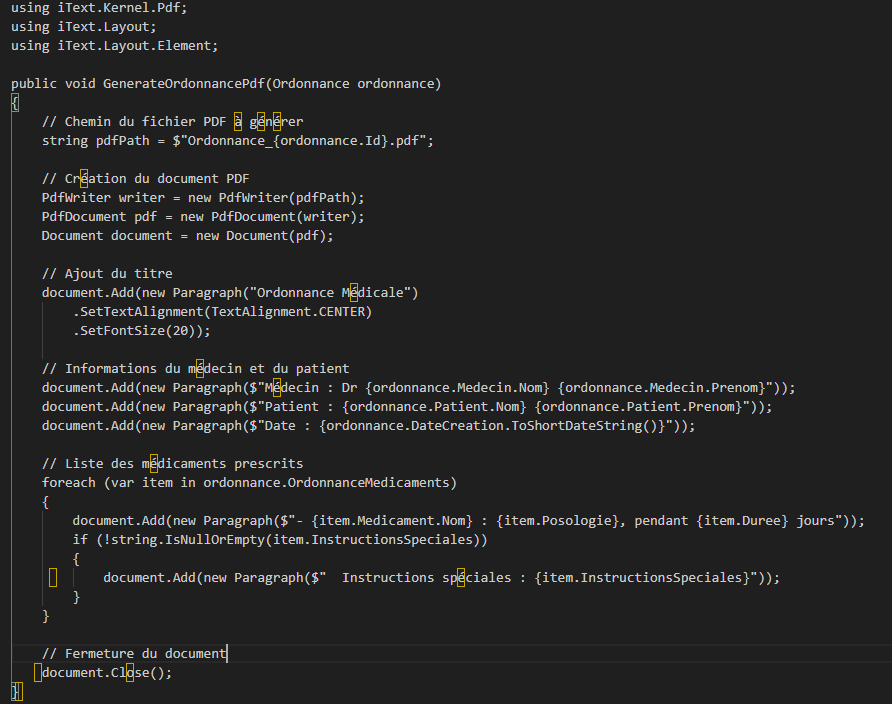
* **Extension aux autres professionnels de santé** : Adapter l'application pour qu'elle soit utilisée par d'autres praticiens, tels que les pharmaciens ou les infirmiers.
* **Développement d'une application mobile** : Créer une version mobile de MedManager pour une utilisation plus flexible en déplacement.
* **Intégration avec les systèmes externes** : Connecter l'application avec d'autres systèmes d'information hospitaliers ou des bases de données nationales pour enrichir les fonctionnalités.
* **Intelligence artificielle** : Incorporer des outils d'aide à la décision basés sur l'IA pour proposer des recommandations personnalisées aux médecins.

En conclusion, le projet **MedManager** représente une avancée significative dans la digitalisation des pratiques médicales au sein de GSB. Il contribue à améliorer la qualité des soins prodigués aux patients tout en facilitant le travail quotidien des médecins. Les retours positifs des utilisateurs confirment la pertinence de l'application et encouragent à poursuivre son développement.

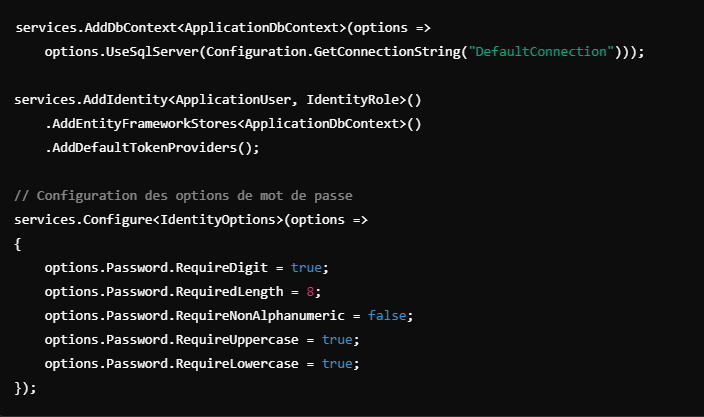
**Annexes**

## **Extraits de code importants**

### **Utilisation de iText7 pour la génération de PDF**



### **Configuration de l'authentification avec Identity**



**Bibliographie**

* **Documentation ASP.NET MVC** :<https://docs.microsoft.com/fr-fr/aspnet/mvc>
* **Guide d'utilisation de Entity Framework** :<https://docs.microsoft.com/fr-fr/ef/>
* **Manuel de iText7** : https://itextpdf.com/en/resources/books/itext-7-building-blocks-pdf-2nd-edition
* **Réglementation RGPD** : https://www.cnil.fr/fr/comprendre-le-rgpd