

# Mini-projet : Application de Suivi de Traitements Médicaux

## Objectif

Créer une application JavaFX permettant à un utilisateur de gérer le suivi de traitements médicaux. L'application doit permettre de :

- Ajouter, modifier et supprimer des patients.
- Gérer les traitements associés à chaque patient.
- Visualiser les informations sur les patients et leurs traitements.

## Contrôles JavaFX à Utiliser

Voici les contrôles JavaFX à inclure, accompagnés d'exemples d'utilisation dans ce projet :

1. **TextField** : pour entrer le nom du patient, le nom du traitement, la posologie, etc.
2. **TextArea** : pour des descriptions plus longues comme les effets secondaires ou les observations médicales.
3. **Button** : pour les actions comme ajouter, modifier, ou supprimer un patient ou un traitement.
4. **Label** : pour afficher les titres, les instructions, ou les champs descriptifs.
5. **RadioButton & ToggleGroup** : pour sélectionner le sexe du patient (Homme / Femme).
6. **CheckBox** : pour marquer si le traitement est actif ou si le patient est sous surveillance particulière.
7. **ComboBox** : pour choisir le type de traitement (Antibiotique, Antalgique, etc.).
8. **ListView** : pour visualiser la liste des patients ou des traitements.
9. **TableView** : pour afficher les données des patients (nom, âge, traitement, état, etc.) dans un tableau.
10. **DatePicker** : pour la date de début/fin de traitement ou la date de naissance du patient.
11. **Slider** : pour ajuster la durée estimée d'un traitement.
12. **Spinner** : pour sélectionner le nombre de prises par jour.
13. **ProgressBar / ProgressIndicator** : pour indiquer l'état d'avancement d'un traitement ou d'une tâche.

14. **ToolTip** : pour afficher des informations supplémentaires en survolant certains éléments.
15. **Menu & MenuBar** : pour structurer les menus comme "Fichier", "Patients", "Aide", etc.
16. **Alert & Dialog** : pour afficher des messages de confirmation ou d'erreur (ex : confirmation de suppression).
17. **Accordion & TitledPane** : pour organiser les informations des patients en sections déroulantes.
18. **ColorPicker** : pour personnaliser l'interface (optionnel).
19. **FileChooser** : pour importer/exporter les données patients ou traitements (CSV, PDF).

### Fonctionnalités à Inclure

1. **Gestion des Patients** : formulaire d'ajout, modification, suppression.
2. **Gestion des Traitements** : création, édition, et suppression de traitements pour chaque patient.
3. **Tableau Récapitulatif** : TableView listant les patients avec leurs traitements.
4. **Recherche & Filtrage** : barre de recherche, filtres par type de traitement ou état du patient.
5. **Statistiques** : nombre de patients, répartition par type de traitement, etc.

### Structure de l'Interface

L'interface sera divisée en sections dédiées :

- Gestion des patients
- Gestion des traitements
- Affichage des statistiques
- Paramètres de l'application

Chaque section devra être intuitive et exploiter les contrôles JavaFX listés ci-dessus.

### Étapes du Projet : Application de Suivi de Traitements Médicaux

#### 1. Définition des Objectifs et Exigences

- **Objectif principal** : permettre aux utilisateurs de suivre les traitements médicaux des patients.
- **Exigences techniques** : utilisation de JavaFX, base de données SQLite, interface interactive avec contrôles JavaFX.

## 2. Conception de la Base de Données

- **Type de base de données**
- Création des diagrammes UML
- Création de la structure de projet

## 3. Création de la Vue (Frontend) avec JavaFX

- Utiliser VBox, HBox, GridPane, ou BorderPane.
- Intégrer les contrôles : TextField, ComboBox, DatePicker, CheckBox, TableView, etc.

## 4. Développement de la Logique Métier (Backend)

- Créer les classes :
  - Patient.java
  - Traitement.java
- Implémenter les méthodes CRUD pour chaque entité.
- ...

## 5. Liaison avec la Base de Données

- Connecter les actions de la vue avec les méthodes de la base via JDBC.
- Fichier Database.java pour la connexion et les requêtes SQL.

## 6. Notifications et Dialogues

- Utiliser Alert pour confirmer les actions et signaler les erreurs.
- Tooltip pour aider à la saisie dans les formulaires.

---

## 7. Tests et Débogage

- Vérifier toutes les actions CRUD.
- Tester les scénarios d'usage typiques (ex : ajout d'un traitement sans patient, etc.).

## 8. Améliorations Supplémentaires

- Ajout de statistiques (ex. traitements en cours, patients sous surveillance).
- Export CSV/PDF des patients et traitements.
- Personnalisation de l'interface avec ColorPicker.

## Structure Finalisée du Projet

### 1. Modèles :

- Patient.java
- Traitement.java

### 2. Base de Données :

- Database.java pour les opérations CRUD.

### 3. Vue :

- MainView.java pour l'interface JavaFX principale.

### 4. Application Principale :

- MedicalTreatmentApp.java pour démarrer l'application.