

Université de la science et de la technologie Houari-Boumediene

Faculté Informatique, Département d'Informatique



Rapport de Mini projet de TP Programmation Orienté Objet.

–Programmation Java.

Gestion de Cabinet Médical : Une Application Java Complète

Réalisé par : Yousfi Mahdi

Saidani Meriem

Aider Abderaouf

Aissat Rania

-Chargée de cours : Mme Bouziane

-Chargée de TD : Mme Bouziane

-Chargé de TP : Mr Ait-Mehdi

Année : 2023/2024.

Introduction :

Le présent rapport décrit le fonctionnement d'une application Java conçue pour la gestion d'un cabinet médical. Cette application a été développée pour faciliter la gestion des dossiers médicaux des patients, la planification des rendez-vous et le suivi des consultations médicales. Elle offre une solution pratique et organisée pour les professionnels de la santé désireux d'optimiser leurs processus de travail.

L'application est structurée en plusieurs classes Java, chacune représentant une entité ou une fonctionnalité spécifique du système. Les principales classes incluent Patient, Consultation, DossierMedical, RendezVous et GestionCabinetMedical. Ce rapport détaillera les responsabilités et les méthodes de chaque classe, ainsi que leur interaction pour atteindre les objectifs de l'application.

Classes :

- **Consultation** : Cette classe représente une consultation médicale. Elle contient des informations telles que la date, l'observation, l'ordonnance et le certificat médical.
- **DossierMedical** : Cette classe représente le dossier médical d'un patient. Elle contient les informations du patient ainsi qu'une liste de consultations effectuées pour ce patient.
- **Patient** : Cette classe représente un patient avec ses informations personnelles telles que le nom, le prénom, le numéro de téléphone, les antécédents médicaux et chirurgicaux, le poids, la taille et les allergies.
- **RendezVous** : Cette classe représente un rendez-vous avec un patient et une date spécifique.
- **GestionCabinetMedical** : Cette classe est le cœur de l'application. Elle gère les patients, les dossiers médicaux, les rendez-vous et fournit des méthodes pour ajouter, rechercher, mettre à jour et supprimer ces éléments.
- **Main** : Cette classe contient la méthode main qui est le point d'entrée de l'application. Elle affiche un menu permettant à l'utilisateur d'interagir avec le système de gestion du cabinet médical.

Opérations de l'application :

- Ajouter un nouveau patient
- Ajouter un rendez-vous pour un patient existant
- Ajouter une nouvelle consultation pour un patient existant
- Afficher la liste des patients
- Afficher la liste des dossiers médicaux
- Afficher la liste des rendez-vous
- Mettre à jour les informations d'un patient existant
- Supprimer un patient existant

Méthodes :

- Classe Consultation :

- Consultation() : Constructeur pour créer une nouvelle instance de Consultation avec les paramètres donnés.
- getDate() : Retourne la date de la consultation.
- getObservation() : Retourne l'observation de la consultation.
- getOrdonnance() : Retourne l'ordonnance de la consultation.
- getCertificatMedical() : Retourne le certificat médical de la consultation.

- Classe DossierMedical :

- DossierMedical() : Constructeur pour créer une nouvelle instance de DossierMedical avec un patient et une liste de consultations.
- ajouterConsultation() : Ajoute une nouvelle consultation à la liste des consultations du dossier médical.
- getPatient() : Retourne le patient associé au dossier médical.
- getConsultations() : Retourne la liste des consultations du dossier médical.
- afficherDossierMedical() : Affiche les informations du dossier médical, y compris les détails du patient et de chaque consultation.

- Classe Patient :

- Patient() : Constructeur pour créer une nouvelle instance de Patient avec les informations fournies.
- Getters pour accéder aux différentes propriétés du patient (getNom(), getPrenom(), getTelephone(), getAntecedentsMedicaux(), getAntecedentsChirurgicaux(), getPoids(), getTaille(), getAllergies()).

- Classe Rendez-Vous :

- RendezVous() : Constructeur pour créer une nouvelle instance de RendezVous avec un patient et une date donnés.
- getPatient() : Retourne le patient associé au rendez-vous.
- getDate() : Retourne la date du rendez-vous.

- Classe GestionCabinetMedical :

- ajouterPatient() : Permet à l'utilisateur de saisir les informations d'un nouveau patient et l'ajoute à la liste des patients et à la liste des dossiers médicaux.
- rechercherPatient() : Recherche et retourne un patient dans la liste des patients à partir de son nom et prénom.
- mettreAJourPatient() : Met à jour les informations d'un patient existant dans la liste des patients.
- supprimerPatient() : Supprime un patient de la liste des patients et son dossier médical de la liste des dossiers médicaux.
- ajouterRendezVous() : Permet à l'utilisateur de saisir un rendez-vous pour un patient existant et l'ajoute à la liste des rendez-vous.
- ajouterConsultation() : Permet à l'utilisateur de saisir une consultation pour un patient existant et l'ajoute au dossier médical du patient.
- rechercherDossierMedical() : Recherche et retourne le dossier médical d'un patient dans la liste des dossiers médicaux.

- *afficherDossiersMedicaux()* : Affiche les informations de tous les dossiers médicaux.
- *afficherRendezVous()* : Affiche la liste de tous les rendez-vous.
- *parseDate()* : Convertit une chaîne de caractères représentant une date au format "dd/MM/yyyy HH:mm" en un objet `Date`.
- *ajouterPatient()* : Ajoute un patient existant à la liste des patients et crée un nouveau dossier médical pour lui.
- *ajouterRendezVous()* : Ajoute un rendez-vous existant à la liste des rendez-vous.

- Classe Main :

- *main(String[] args)* : Méthode principale qui exécute le programme. Elle crée une instance de `GestionCabinetMedical`, affiche un menu et effectue les opérations choisies par l'utilisateur.
- *afficherPatients()* : Affiche la liste des patients enregistrés dans le cabinet médical.
- *mettreAJourPatient()* : Permet à l'utilisateur de saisir les informations d'un patient existant et met à jour ses informations.
- *supprimerPatient()* : Permet à l'utilisateur de saisir le nom et le prénom d'un patient existant et le supprime du cabinet médical.
- *saisirInformationsPatient()* : Permet à l'utilisateur de saisir les informations d'un nouveau patient et retourne un objet `Patient` avec ces informations.

EXEMPLES D'EXECUTION :

Menu Principale :

```
Que voulez-vous faire ?
1. Ajouter un patient
2. Ajouter un rendez-vous
3. Ajouter une consultation
4. Afficher la liste des patients
5. Afficher la liste des dossiers médicaux
6. Afficher la liste des rendez-vous
7. Mettre à jour un patient
8. Supprimer un patient
9. Quitter
Votre choix :
```

Ajout

```
Entrez le nom du patient :
Yousfi
Entrez le prénom du patient :
Mahdi
Entrez le numéro de téléphone du patient :
054789312
Entrez les antécédents médicaux du patient (ou 'aucun') :
aucun
Entrez les antécédents chirurgicaux du patient (ou 'aucun') :
aucun
Entrez le poids du patient :
75
Entrez la taille du patient :
177
Entrez les allergies du patient (séparées par des virgules) :
aucun
Patient ajouté avec succès.
```

Ajout d'un patient :

d'un rendez-vous :

```

Que voulez-vous faire ?
1. Ajouter un patient
2. Ajouter un rendez-vous
3. Ajouter une consultation
4. Afficher la liste des patients
5. Afficher la liste des dossiers médicaux
6. Afficher la liste des rendez-vous
7. Mettre à jour un patient
8. Supprimer un patient
9. Quitter
Votre choix : 2
Entrez le nom du patient :
Yousfi
Entrez le prénom du patient :
Mahdi
Entrez la date du rendez-vous (format : dd/MM/yyyy HH:mm) :
Que voulez-vous faire ?
1. Ajouter un patient
2. Ajouter un rendez-vous
3. Ajouter une consultation
4. Afficher la liste des patients
5. Afficher la liste des dossiers médicaux
6. Afficher la liste des rendez-vous
7. Mettre à jour un patient
8. Supprimer un patient
9. Quitter
Votre choix : 3
Entrez le nom du patient :
Yousfi
Entrez le prénom du patient :
Mahdi
Entrez la date de la consultation (format : dd/MM/yyyy HH:mm) :
22/08/2024 15:30
Entrez l'observatoire :
Contrôle des analyses
Entrez l'ordonnance :
Doliprane
Entrez le certificat médical (ou 'aucun') :
aucun

```

Ajout d'une consultation :

Affichage des rendez-vous :

Interface Graphique (Indépendante) :

C'est un code JavaFX qui permet à l'utilisateur de saisir les informations d'un patient et de prendre des rendez-vous. L'application comporte deux interfaces principales :

- Fiche Patient

- Cette interface permet de saisir les détails du patient, tels que le nom, le prénom, la date de naissance, le numéro de téléphone, l'adresse, la pathologie, le traitement actuel, le poids, la taille, les allergies et les chirurgies.

- Les informations sont affichées dans une grille et peuvent être sauvegardées en cliquant sur le bouton "Sauvegarder".

- La méthode *sauvegarderFichePatient* est actuellement une simple implémentation de démonstration qui affiche les informations saisies dans la console.

- Rendez-Vous

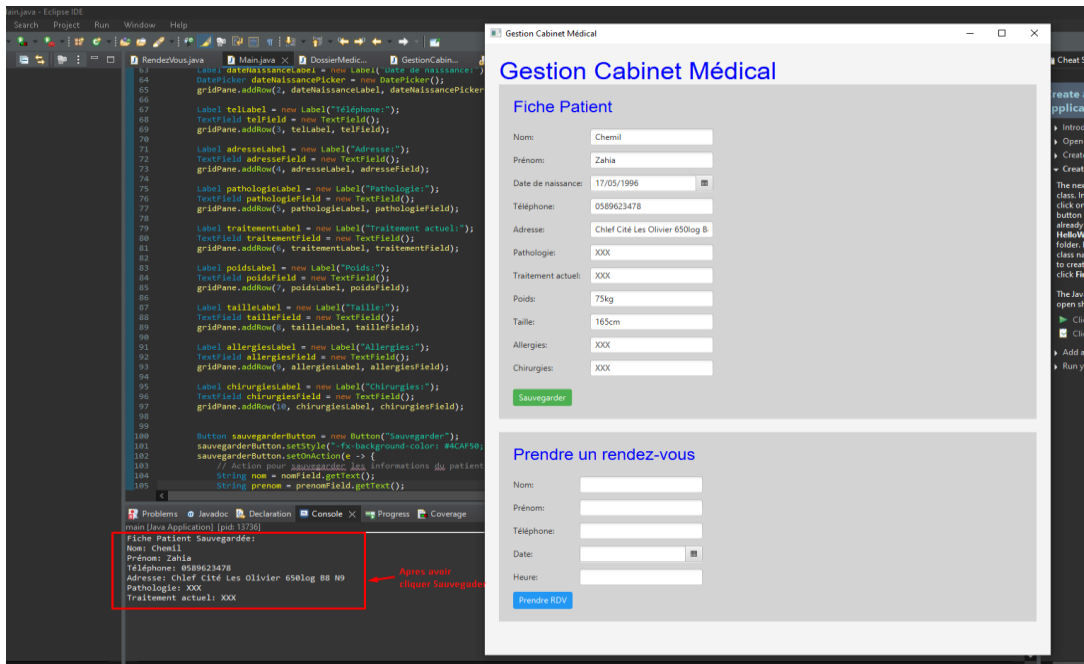
- Cette interface permet de prendre un rendez-vous en saisissant le nom, le prénom, le numéro de téléphone du patient, ainsi que la date et l'heure du rendez-vous.

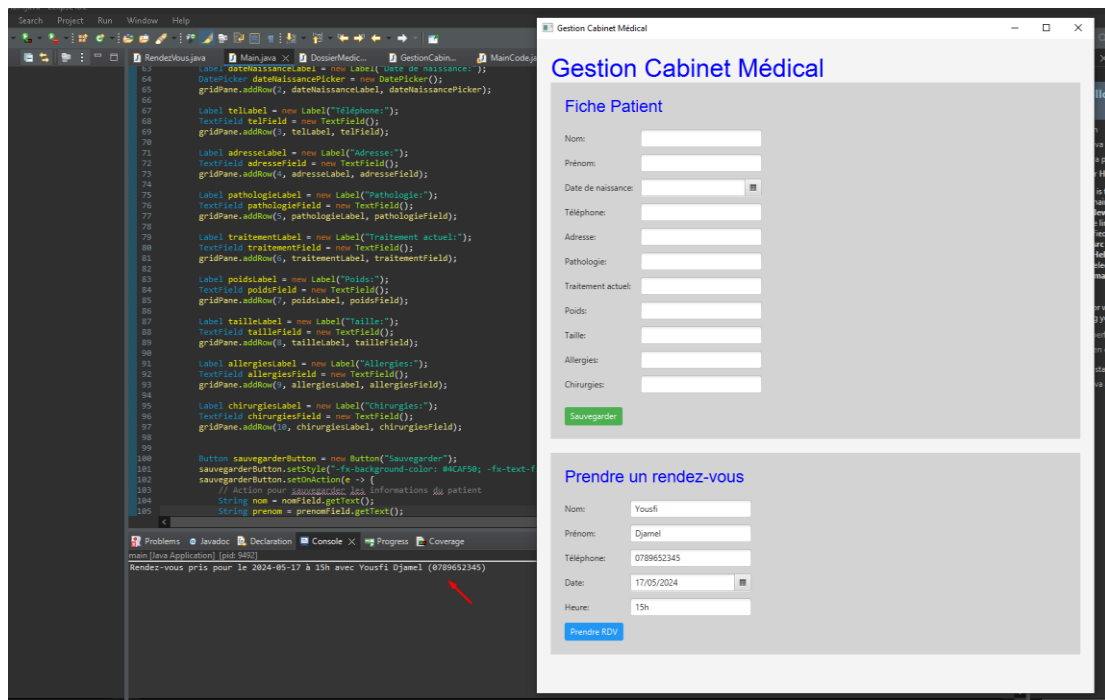
- Après avoir rempli les champs, l'utilisateur peut cliquer sur le bouton "Prendre RDV" pour confirmer le rendez-vous.

- La méthode `prendreRendezVous` de la classe `CabinetMedical` (non présente dans le code fourni) devrait être appelée pour enregistrer le rendez-vous.

The screenshot displays a web application window titled "Gestion Cabinet Médical". The interface is divided into two main sections. The top section, titled "Fiche Patient", contains a form with the following fields: "Nom:", "Prénom:", "Date de naissance:" (with a calendar icon), "Téléphone:", "Adresse:", "Pathologie:", "Traitement actuel:", "Poids:", "Taille:", "Allergies:", and "Chirurgies:". A green "Sauvegarder" button is located at the bottom of this section. The bottom section, titled "Prendre un rendez-vous", contains a form with the following fields: "Nom:", "Prénom:", "Téléphone:", "Date:" (with a calendar icon), and "Heure:". A blue "Prendre RDV" button is located at the bottom of this section.

EXEMPLES D'UTILISATION :





Conclusion :

En résumé, l'application de gestion du cabinet médical décrite dans ce rapport offre une solution complète pour la gestion des patients, des dossiers médicaux, des rendez-vous et des

consultations. Grâce à sa conception modulaire en classes Java, elle permet une maintenance et une évolution faciles pour répondre aux besoins futurs.

En fin de compte, cette application offre une base solide pour la gestion efficace des tâches quotidiennes dans un cabinet médical, en fournissant une structure organisée et des fonctionnalités clés pour les professionnels de la santé.