



Systeme de Gestion Médicale en java





SQUELETTE DU RAPPORT

1. Introduction

- Présentation du problème :
- Objectif principal :
- Avantages :

2. Analyse des Besoins

2.1 Besoins Fonctionnels

2.2 Besoins Non Fonctionnels

3. Conception

- Cas d'utilisation : fonctionnalité de chaque profil.

- Classes : description, détail

4. Développement

- Technologies utilisées :
- Description des API principales.

5. Fonctionnalités

5.1 Gestion des Dossiers Médicaux

5.2 Gestion des Rendez-vous

5.3 Gestion des Fiches Patients

5.4 Génération de Documents

7. Conclusion et Perspectives

- Résumé des bénéfices apportés par le logiciel.
- Améliorations futures possibles

•Ce projet est fait par les étudiants:

Bendjama ines 222231368803

Souai ines 232331365609

1. Introduction :

- Présentation du problème :

La gestion d'un cabinet médical devient complexe avec l'augmentation du nombre de patients. La prise de rendez-vous, le suivi des traitements et la gestion des dossiers nécessitent beaucoup de temps et peuvent entraîner des erreurs ou des retards. De plus, les exigences en matière de confidentialité des données et la charge administrative peuvent nuire à l'efficacité du cabinet, affectant ainsi la qualité des soins et la satisfaction des patients.

- Objectif principal :

L'objectif principal est d'optimiser la gestion du cabinet médical en automatisant les tâches administratives, améliorant ainsi l'efficacité des processus, la gestion des dossiers patients et la prise de rendez-vous. Cela permettra de réduire les erreurs, d'améliorer la satisfaction des patients et de libérer du temps pour que les professionnels de santé puissent se concentrer davantage sur les soins.

- Avantages :

Gain de temps : L'automatisation des tâches administratives permet de réduire la charge de travail, libérant ainsi du temps pour les professionnels de santé.

Réduction des erreurs : La gestion électronique des dossiers et des rendez-vous minimise les risques d'erreurs humaines.

Amélioration de la satisfaction des patients : Des rendez-vous plus fluides et un suivi plus précis contribuent à une meilleure expérience pour les patients.

Meilleure gestion des données : L'organisation numérique des dossiers garantit la sécurité, la confidentialité et la facilité d'accès aux informations médicales.

Optimisation des ressources : Une gestion plus efficace permet de mieux répartir les ressources et de réduire les coûts opérationnels.

2. Analyse des Besoins :

2.1 Besoins Fonctionnels :

- **Gestion des fiches médicaux :** Le système doit permettre la Création, la mise à jour, la consultation et l'archivage des fiches médicaux des patients. Ces fiches doivent inclure des informations personnelles, historiques médicaux, résultats de tests, traitements prescrits, etc.

- **Gestion des rendez-vous :** Le système doit gérer les rendez-vous des patients, en permettant leur prise en charge en ligne, la planification, ainsi que l'envoi de rappels automatiques. Les médecins doivent également pouvoir voir leur emploi du temps et ajuster les disponibilités en fonction des besoins.

- **Gestion des fiches patients :** Les fiches patients doivent contenir des informations détaillées sur chaque individu : données personnelles, antécédents médicaux, traitements en cours, etc. Le système doit permettre une recherche rapide et une mise à jour facile de ces informations.
- **Génération automatique des certificats médicaux et ordonnances :** Le système doit générer automatiquement des certificats médicaux et des ordonnances après chaque consultation, avec des champs pré-remplis basés sur les informations enregistrées dans le dossier du patient. Cela permet de gagner du temps et d'éviter les erreurs.

2.2 Besoins Non Fonctionnels :

Les besoins non fonctionnels se concentrent sur l'expérience utilisateur et la performance du système, sans traiter directement des fonctionnalités spécifiques. Deux besoins principaux sont identifiés :

- **Interface utilisateur intuitive :** L'interface doit être simple, claire, et facile à utiliser, même pour les utilisateurs non expérimentés. Elle doit permettre une navigation fluide avec des éléments bien organisés, des retours visuels clairs et une hiérarchisation logique des informations. L'interface doit être adaptable aux préférences des utilisateurs (langue, taille du texte, etc.).
- **Accessibilité multi-plateforme (web, mobile) :** Le système doit être accessible sur plusieurs appareils (ordinateur, tablette, smartphone), avec une expérience cohérente et une synchronisation des données entre toutes les plateformes, tout en garantissant la sécurité des informations.

3. Conception :

3.1 Cas d'utilisation : fonctionnalité de chaque profil

1. La première chose que l'utilisateur voit lorsqu'il ouvre le programme est la page de Login :



Comme vous pouvez le voir sur cette page, vous devez choisir votre statut et entrer votre nom d'utilisateur ainsi que votre mot de passe. Ensuite, cliquez sur Connexion pour accéder, ou sur Réinitialiser pour réinitialiser les champs.

Selon le statut que vous sélectionnerez, une nouvelle page s'affichera. Ainsi, si vous choisissez :

- **Admin :**
L'admin a beaucoup de fonctions à accomplir :

1. La gestion des employés :

The screenshot shows a web application titled 'GESTION DES EMPLOYES'. It features a navigation bar with links: 'Employés', 'Patients', 'Reservations', 'Feedback', 'Voir Statistique', and 'avis patients'. The main content area is divided into two sections. On the left, there is a form for adding or editing an employee with fields for 'ID', 'Nom', 'Prénom', 'Secteur' (a dropdown menu currently showing 'Medical'), 'Spécialité', 'Ancienneté', 'Username', and 'Password'. Below the form are buttons for 'Ajouter', 'Supprimer', 'Modifier', and 'RESET'. On the right, there is a table titled 'La liste de mes employées' displaying a list of employees with columns for ID, Nom, Prénom, Secteur, Spécialité, Ancienneté, Login, and Password.

ID	Nom	Prénom	Secteur	Spécialité	Ancienneté	Login	Password
1	zayd	saadi	Médical	Cardiologie	20	saadi-zaid	password123
2	houssam	démouche	Paramédical		8	hou_démouche	acadegp3
3	Smith	Alice	Medical	Cardiology	5	alice_smith	password123
4	Johnson	Bob	Surgical	Orthopedic Sur	9	bob_johnson	securepass
5	Smith	Alice	Medical	Cardiology	5	alice_smith	password123
6	abdou	djoudi	Médical	généraliste	20	abdou.djoudi	1234

. L'admin aura la possibilité de supprimer, ajouter ou modifier un employé, en saisissant les informations nécessaires, telles que le Nom, le Prénom, le Secteur, la Spécialité, Ancienneté, Username, Password, ID. La case » RESET « POUR Réinitialiser les champs. Dans la section de droite, un tableau affiche les informations des employés ajoutés, incluant leur ID, nom, prénom ...etc. Si l'admin souhaite rechercher un employé, il saisit son ID dans le champ « ID » et clique sur "Entrée", puis le système effectuera la recherche automatiquement.

2. Gestion des patients :

3.

The screenshot shows a web application titled 'GESTION DES PATIENTS'. It features a navigation bar with links: 'Employés', 'Patients', 'Reservations', 'Feedback', 'Voir Statistique', and 'avis patients'. The main content area is divided into two sections. On the left, there is a form for adding or editing a patient with fields for 'NSS', 'Nom', 'Prénom', 'Date de naissance', 'Adresse', 'Téléphone', and 'Mail'. Below the form are buttons for 'Ajouter', 'Supprimer', 'Modifier', and 'RESET'. On the right, there is a table titled 'les patients de ma clinique' displaying a list of patients with columns for NSS, Nom, Prénom, Date de naissance, Adresse, Téléphone, and Email.

NSS	Nom	Prénom	Date de naissance	Adresse	Téléphone	Email
123456789	touma	bilal	2000-01-01	chetyaa	0540940470	john@example.c...
123456789	bilal	touma	2000-01-01	chetyaa	0540940470	john@example.c...
987654321	kadi	dina	2003-05-10	mila	0665458912	dina@gmail.com
987654322	djoudi	abdou	2003-10-06	456 touggert	987-654-3210	jane@example.c...
987654323	mouh	lab	2002-10-27	456 tatif	987-654-3217	jane@example.c...

. L'admin aura la possibilité de supprimer, ajouter ou modifier un patient, en saisissant les informations nécessaires, telles que le Nom, le Prénom, NSS (Numéro de Sécurité Social), l'Adresse, Téléphone, Mail. RESET POUR Réinitialiser les champs.

Dans la section de droite, un tableau affiche les informations des patients ajoutés, incluant leur NSS, nom, prénometc. Si l'admin souhaite rechercher un patient, il saisit son NSS dans le champ NSS et clique sur "Entrée", puis le système effectuera la recherche automatiquement.

4. Gestion des Reservation :

Employés
Patients
Reservations
Feedback
Voir Statistique
avis patients

GESTION DES RESERVATIONS

Numero Reservation : 8
NSS : 987654321
Date : 2024-04-27
Heure : 00:00
Type : Paramédical
Service : Orthopedic Surgery
Etat : Non confirmée

Ajouter
Annuler
Modifier
RESET

num_res	NSS	Date	Heure	Type	Service	Etat
1	123456789	2024-04-25	00:00	Paramédical	Cardiology	1
6	987654323	2024-04-25	00:00	Medical	change	1
7	123456789	2024-04-25	00:00	New Patient	Cardiology	1
8	987654321	2024-04-27	00:00	Follow-up	Orthopedic Surg.	0
9	123456789	2024-04-25	00:00	New Patient	Cardiology	1
10	987654321	2024-04-27	00:00	Follow-up	Orthopedic Surg.	0
12	123456788	2024-04-25	00:00	Paramédical	Cardiology	0
13	123456789	2024-04-25	00:00	Paramédical	Cardiology	0
14	123456789	2024-04-25	00:00	Paramédical	Cardiology	1
15	123456789	2024-04-25	00:00	Paramédical	Cardiology	0

. L'admin a le droit d'ajouter une nouvelle réservation, de l'Annuler ou de la Modifier, en saisissant les informations nécessaires, telles que NSS (Numéro de Sécurité Social), la Date, l'Heure, le Type, le Service et l'Etat. » Numero Reservation « est généré automatiquement et chaque réservation a un numéro distinct. RESET POUR Réinitialiser les champs. Dans la section de droite, un tableau affiche les réservations disponibles, incluant leur num_res, NSS, dateetc. Si l'admin souhaite rechercher une réservation, il saisit NSS de patient qui a la réservation dans le champ NSS et clique sur "Entrée", puis le système effectuera la recherche automatiquement.

• Médecin/Infirmier :

Si l'utilisateur sélectionne le statut 'MedecinInfirmier' Il a le droit d'accéder à cette page :

Informations

CONSULTATION

Numero consultation : 15
nss : 123456789
Date de consultation: le 2024-04-25 à 00:00
Service : Cardiology

Traitement :

allergie

Etat :

grave

Supprimer
Terminer

Consultations a faire

num_consultation	NSS	Date	Heure	Service
15	123456789	2024-04-25	00:00	Cardiology
17	123456789	2024-04-25	00:00	Cardiology

Cette page permet les infirmiers et les médecins de consulter le nombre de consultations à réaliser, ainsi que le num_consultation, les numéros NSS des patients, les dates des consultations, les services et les traitements. Toutes ces informations seront affichées dans un tableau dans la section de droite. Si Le médecin ou l’infirmier souhaite rechercher la consultation d’un patient, il doit saisir le NSS du patient dans le champ NSS, puis clique sur "Entrée". Les résultats s'affichent dans la section de droite, incluant le numéro de consultation, le nss du patient, la date de consultation, le service et le traitement.

- Chef médecin :

Si l'utilisateur sélectionne le statut 'MedecinInfirmier' Il a le droit d'accéder à cette page :

num_res	NSS	Date	Heure	Type	Service	Etat
1	123456789	2024-04-25	00:00	Paramédical	Cardiology	1
6	987654323	2024-04-25	00:00	Médical	chirurgie	1
7	123456789	2024-04-25	00:00	New Patient	Cardiology	1
9	123456789	2024-04-25	00:00		Cardiology	1
14	123456789	2024-04-25	00:00	Paramédical	Cardiology	1

Cette page permet le chef médecin de sélectionner une réservation dans le tableau des consultations situé à gauche. Une fois la réservation sélectionnée, elle apparaît automatiquement dans la section de droite. Ensuite, pour l'affecter à un infirmier ou à un médecin, l'utilisateur doit cliquer sur la case correspondante, "Infirmier" ou "Médecin".

- Patient :

Si l'utilisateur sélectionne le statut 'Patient' il a le droit d'accéder à cette page

numero_c...	NSS	Date	service	code_empl...
1	123456789	2024-04-28	Cardiology	1
3	123456789	2024-04-28	Cardiology	2
5	123456789	2024-04-28	Cardiology	5
7	123456789	2024-04-25	Cardiology	1
12	123456789	2024-04-25	Cardiology	1
13	123456789	2024-04-25	Cardiology	2
14	123456789	2024-04-25	Cardiology	5
15	123456789	2024-04-25	Cardiology	1
16	123456789	2024-04-25	Cardiology	3
17	123456789	2024-04-25	Cardiology	1
18	123456789	2024-04-25	Cardiology	1

numero_consult...	code_employe	traitement	etat
7	1	ffff	ffff
12	1	allergie	grave
15	1	allergie	grave
15	1		hhhhh
17	1	dfdd	ddd
18	1	good	good

Sur cette page, le patient peut consulter ses réservations, avec les détails suivants : la date et l'heure, le numéro de réservation, le service, le NSS, le type de réservation et son état.

3•2 Classes : description, détail.....

1. **Class Login()** : Cette classe gère l'authentification de l'utilisateur en vérifiant le nom d'utilisateur et le mot de passe par rapport aux informations stockées dans une base de données et accorde l'accès si les informations sont correctes.

2. **Class Admin()** :

- **Méthodes :**

- **Update_table_patient()** : La méthode `Update_table_patient()` vide d'abord la table existante, puis ajoute trois lignes de données fictives représentant des patients, telles que leur NSS, nom, prénom, adresse, téléphone et email, pour afficher ces informations dans la table.
- **clear_patient()** : Cette fonction vide les champs de texte utilisés pour saisir les informations du patient.
- **nbrvisitePat()** : Cette fonction récupère le nombre de consultations d'un patient et met à jour le label `nbVisitesPat`. En cas d'erreur, elle affiche le message "Le patient n'existe pas ou n'a pas de visites."
- **clear_employe()** : Cette fonction vide les champs de saisie et réinitialise la sélection du secteur pour un employé.
- **Update_table_employe()** La méthode `Update_table_employe()` met à jour le tableau des employés en supprimant les anciennes données et en ajoutant des données fictives pour les employés.
- **nbrvisiteEmp()** : Cette fonction récupère le nombre de consultations d'un employé et met à jour le label `nbVisites` avec ce nombre. En cas d'erreur, elle affiche un message indiquant que l'employé n'existe pas ou n'a pas de visites.
- **clear_rservaion()** : Cette fonction réinitialise les champs de saisie des informations de réservation à leurs valeurs par défaut.
- **update_Table_Reservation** : La méthode `update_Table_Reservation()` vide d'abord le tableau des réservations, puis ajoute de nouvelles données fictives sur les réservations, telles que des numéros de réservation, des NSS, des dates, des heures, des types de service, des spécialités et des états de confirmation.
- **ResAnnulee()** : Cette fonction récupère le nombre de réservations annulées (ayant l'état 2) depuis la table de donnée et met à jour le label `ResAnul` avec ce nombre.
- **update_Table_consultation()** La méthode `update_Table_consultation()` vide et met à jour le tableau des consultations avec des données fictives sur les consultations, incluant des informations telles que le numéro de consultation, le NSS, la date, l'heure et la spécialité.
- **clear_avis()** : Cette fonction réinitialise les champs de saisie des avis à leurs valeurs par défaut.

3. **Class Patient :**

- **Méthodes :**

- **initComponents()** :

La méthode `initComponents()` initialise les composants graphiques de la fenêtre, en définissant une table de réservations avec des colonnes spécifiques, et en affichant un titre. Elle configure également la mise en page avec un `GroupLayout` pour organiser le placement des composants (comme un tableau de réservations et un label

pour l'intitulé). Cette méthode fait l'initialisation des composants de l'interface graphique, incluant un tableau de réservations dans un JScrollPane et un label indiquant 'Votre réservation'.

- `update_Table_Reservation()` :

La méthode `update_Table_Reservation()` met à jour la table de réservations en ajoutant des données codées en dur pour les réservations, avec des informations comme le numéro de réservation, le NSS, la date, l'heure, le type, le service, l'état et l'observation, puis elle applique ces données à la table à l'aide d'un modèle (`DefaultTableModel`).

4. Class ChefMedecin() :

- Méthodes :

- `update_Table_Reservation()` :

Cette fonction récupère les réservations avec l'état "1" depuis la table de données, efface les anciennes lignes du tableau et ajoute les nouvelles. En cas d'erreur, elle affiche le message d'erreur.

- `insertPatient()` :

La méthode `insertPatient()` permet d'ajouter les informations d'un patient, de vérifier que tous les champs sont remplis, d'afficher un message de succès et de vider les champs du formulaire après l'insertion.

- `clearPatientFields()` : La méthode `clearPatientFields()` vide tous les champs de saisie du formulaire en réinitialisant leur contenu à une chaîne vide.

5. Class MedecinInfirmier():

- Méthodes :

- `update_Table_consultation ()` :

La méthode `update_Table_Reservation()` extrait les informations d'une ligne sélectionnée dans le tableau de réservations et les enregistre dans des variables pour une utilisation ultérieure.

- `clearPatientFields()` :

Cette fonction réinitialise les champs de saisie du traitement et de l'état de la consultation.

- `insertPatient()` : La méthode `insertPatient()` récupère les informations d'un patient depuis les champs de texte, vérifie qu'aucun champ n'est vide, simule l'insertion dans une base de données, affiche un message de confirmation, puis vide les champs après l'insertion.

6.class EmployeeList :

La méthode `EmployeeList()` est un constructeur d'une fenêtre graphique qui affiche une liste d'employés (médecins ou infirmiers) dans un panneau déroulant. Elle permet de choisir un employé en cliquant sur un bouton "Choisir", ce qui affiche un message de confirmation. Les employés sont définis par des données fictives, et la fenêtre adapte son titre et ses données selon le type d'employé (médecin ou infirmier). Elle utilise un `JScrollPane` pour permettre le défilement et dispose les informations des employés dans un `JPanel` avec des étiquettes et un bouton d'action.

4. Développement :

- Technologies utilisées :
 - Front-end :

L'interface graphique a été développée avec [Java Swing](#), des bibliothèques Java permettant de créer des interfaces utilisateur interactives et modernes.

- Back-end :

Le back-end est développé en Java, avec l'environnement [NetBeans](#) pour le codage et le débogage.

- Les table de données :

patient	Employe	reservation	consultation	avis
nss	code_employe	Numero_reservation	Numero_consultation	nss
Non	Nom	Nss	Nss	commentaire
Prenom	Prenom	Date_reservation	Date_reservation	note
Adresse	Secteur	Type	Type	Numero_consue
Telephone	Specialiste	Service	Service	
email	Enciennete	Etat	Etat	
Datte_naissance	login			
	password			

5. Fonctionnalités :

5.1 Gestion des fiches Médicaux :

- Création, modification, et consultation des fiches:

Lorsqu'un patient consulte un médecin, une nouvelle fiche de consultation médicale est créée et enregistrée dans la table de données. Cette fiche doit contenir plusieurs informations cruciales pour le suivi et le traitement du patient . `update_Table_consultation()` permet de d'afficher toutes les consultations dans un tableau en utilisant des requêtes SQL. Ce tableau contient des informations telles que le numéro de consultation, le NSS du patient, la date et heure de la consultation, ainsi que le service médical

- Historique des consultations :

Chaque consultation est enregistrée et peut être consultée dans le système. La classe `MedecinInfirmier` contient des méthodes pour afficher les consultations passées via la méthode `update_Table_consultation()`. Elle permet d'afficher des détails comme la date, l'heure et le service de chaque consultation.

5.2 Gestion des Rendez-vous :

- Les rendez-vous sont gérés via une interface intuitive :

pour une prise en charge efficace Saisie et gestion des rendez-vous : L'administrateur ou un employé peut créer et gérer des rendez-vous pour les patients. La table des réservations est mise à jour avec les détails des rendez-vous grâce à la méthode `update_Table_Reservation()`.

5.3 Gestion des Fiches Patients :

Fiches contenant les informations personnelles : Lorsqu'un patient se présente, l'administrateur ou le médecin peut enregistrer ou mettre à jour ses informations dans les tables de données. La consultation des fiches médicaux est gérée par la classe `**Admin**`. La méthode `Update_table_patient()` récupère toutes les données des patients depuis les table de données (avec la requête `**SELECT * FROM patient**`) et les affiche dans un tableau (`patient_table`). Cela permet à l'administrateur de consulter tous les dossiers médicaux des patients, y compris les informations personnelles

Historique médical : L'historique des consultations est enregistré dans un tableau (`patient_table`)

5.3 Génération de Documents :

Le système génère automatiquement des documents médicaux comme des ordonnances et des certificats médicaux.

6. Conclusion et Perspectives :

- Résumé des bénéfices apportés par le logiciel :

Le logiciel offre une gestion centralisée des informations médicales et administratives des patients, permettant une consultation rapide et efficace des dossiers médicaux, des rendez-vous et des historiques de consultations. Grâce à une interface intuitive et une gestion automatisée des rendez-vous, le système facilite le travail des médecins, des administrateurs et des employés de l'hôpital, améliorant ainsi la qualité des soins et l'organisation du cabinet ou de l'hôpital.

- Améliorations futures possibles :

- Système de recherche avancée :

Intégration d'une fonction de recherche pour trouver rapidement des informations spécifiques sur les patients ou leurs consultations passées.

- Sécurité des données :

Implémentation de fonctionnalités de sécurité avancées pour protéger les informations sensibles des patients, telles que le chiffrement des données et l'authentification forte.

- Accessibilité mobile :

Développer une version mobile de l'application pour permettre aux utilisateurs d'accéder aux informations et de gérer les rendez-vous depuis leurs smartphones.

- **Intégration avec des pharmacies :** Permettre une synchronisation avec des systèmes de gestion de pharmacie afin que les ordonnances puissent être envoyées directement aux pharmacies pour traitement.