 

***Université des Sciences et de la***

**Faculté d’informatique**

**Rapport sur :**

***Technologie Houari***

***Boumediene***

**Faculté d’informatique**

***Rapport sur :***

Gestion Médical

Rédiger par :

Fegas Lokman abdelhakim

Laid Mohamed

Ferhat Abderrahmane

Kedjour chakib abderrahmane

**1. Introduction**

Ce projet est un système de gestion de cabinet médical conçu pour rationaliser la gestion des dossiers médicaux, des rendez-vous, des consultations et des informations des patients dans un cabinet médical. Le système est construit en utilisant Java avec une interface utilisateur graphique (GUI) créée avec Swing, et il utilise SQLite pour la gestion de la base de données. Le système vise à améliorer l'efficacité et l'organisation dans les cabinets médicaux en fournissant une plateforme intégrée pour gérer toutes les opérations essentielles.

**2. Composants du Projet**

Le projet se compose de plusieurs composants essentiels qui travaillent ensemble pour créer un système de gestion de cabinet médical fonctionnel et efficace. Chaque composant remplit un rôle spécifique dans le fonctionnement global du système.

1. Classes de Base :

* Les classes de base définissent les entités fondamentales du système, telles que les patients, les médecins, les rendez-vous, les consultations et les dossiers médicaux.
* Chaque classe de base comprend des attributs qui capturent les informations pertinentes pour l'entité correspondante, par exemple, nom, adresse, numéro de téléphone, antécédents médicaux, spécialité, etc.
* Ces classes établissent la structure de base sur laquelle repose l'ensemble du système, en décrivant les données et les comportements associés à chaque entité.

1. Classes DAO (Data Access Object):

* Les classes DAO sont responsables de l'interaction avec la base de données du système.
* Chaque classe DAO est associé à une entité spécifique (par exemple, PatientDAO pour les patients, DoctorDAO pour les médecins) et fournit des méthodes pour effectuer des opérations CRUD (Create, Read, Update, Delete) sur les données de cette entité dans la base de données.
* Elles encapsulent la logique d'accès aux données, permettant ainsi au reste du système de fonctionner indépendamment de la structure sous-jacente de la base de données.

1. Classes de Service :

* Les classes de service agissent comme une couche intermédiaire entre l'interface utilisateur (GUI) et les classes DAO.
* Elles fournissent une abstraction des opérations métier et agissent comme des points d'entrée pour l'accès aux fonctionnalités du système.
* Chaque classe de service expose des méthodes pour effectuer des actions spécifiques sur les entités du système, telles que l'ajout, la mise à jour, la suppression et la récupération de données.
* Les services encapsulent la logique métier et coordonnent les interactions entre l'interface utilisateur et la couche d'accès aux données, assurant ainsi la cohérence et la validité des opérations effectuées.

1. Panneaux d'Interface Graphique (GUI) :

* Les panneaux GUI fournissent une interface utilisateur interactive pour interagir avec le système de gestion de cabinet médical.
* Chaque panneau est dédié à une fonctionnalité spécifique du système, telle que la gestion des patients, la planification des rendez-vous, l'enregistrement des consultations, etc.
* Les panneaux GUI permettent aux utilisateurs de visualiser et de manipuler les données du système de manière conviviale, en fournissant des champs de saisie, des boutons d'action et des affichages de données.
* Ils sont conçus pour offrir une expérience utilisateur intuitive et efficace, en facilitant la navigation et l'interaction avec les différentes fonctionnalités du système.

1. Validation des Entrées :

* La validation des entrées garantit que les données saisies par les utilisateurs sont cohérentes et conformes aux exigences du système.
* Cette composante inclut des mécanismes de validation des données au niveau de l'interface utilisateur, tels que des vérifications de format et des contraintes de champ obligatoire.
* La validation des entrées contribue à maintenir l'intégrité des données du système, en évitant les erreurs de saisie et les incohérences qui pourraient compromettre la qualité des informations stockées.

Ensemble, ces composants forment les éléments constitutifs du système de gestion de cabinet médical, fournissant une structure robuste et fonctionnelle pour la gestion efficace des opérations médicales. Chaque composant joue un rôle crucial dans la réalisation des objectifs du projet, en contribuant à la création d'un système complet et bien intégré pour les professionnels de la santé.

**3. Classes de Base**

Les classes de base du projet définissent les entités principales avec lesquelles le système interagit. Chaque classe représente un concept clé dans le domaine de la gestion de cabinet médical et encapsule les données associées ainsi que les comportements spécifiques à cette entité.

1. Classe Personne :

* La classe Personne sert de classe de base pour les entités comme les patients et les médecins.
* Elle contient des attributs communs à toutes les personnes, tels que le nom, l'adresse, le numéro de téléphone, etc.
* Cette classe fournit une structure uniforme pour la gestion des informations personnelles des individus impliqués dans le système.

1. Classe Patient :

* La classe Patient étend la classe Personne et inclut des attributs supplémentaires spécifiques aux patients.
* Parmi ces attributs figurent les antécédents médicaux, les informations sur les assurances, les allergies, etc.
* Elle représente les détails médicaux et personnels des patients, permettant ainsi au système de suivre et de gérer leur historique de santé.

1. Classe Médecin :

* La classe Médecin étend également la classe Personne et contient des attributs propres aux médecins.
* Ces attributs comprennent la spécialité, l'horaire de travail, l'adresse de consultation, etc.
* Elle représente les professionnels de la santé qui fournissent des services médicaux dans le cadre du système et permet de les identifier et de les gérer efficacement.

1. Classe Rendez-vous :

* La classe Rendez-vous représente les rendez-vous programmés entre les patients et les médecins.
* Elle contient des informations telles que la date, l'heure, le patient concerné et le médecin associé.
* Elle facilite la gestion des horaires et des consultations en permettant la planification et le suivi des rendez-vous médicaux.

1. Classe Consultation :

* La classe Consultation enregistre les détails des consultations médicales entre les patients et les médecins.
* Elle comprend des informations telles que le diagnostic, le traitement recommandé, les observations du médecin, etc.
* Elle permet de documenter les interactions médicales et de suivre l'évolution des traitements et des conseils donnés aux patients.

1. Classe Dossier Médical :

* La classe Dossier Médical est responsable de la gestion des dossiers médicaux des patients.
* Elle relie les informations sur les patients aux consultations, aux diagnostics, aux prescriptions, etc., pour former un dossier complet.
* Elle facilite le stockage et la récupération des données médicales et contribue à assurer une prise en charge appropriée des patients.

Chaque classe de base joue un rôle essentiel dans la représentation et la gestion des entités pertinentes pour le système de gestion de cabinet médical. Ensemble, elles fournissent une base solide pour la manipulation des données et la réalisation des fonctionnalités du système, permettant ainsi une gestion efficace des opérations médicales

**4. Classes DAO (Data Access Object)**

Les classes DAO sont chargées de l'interaction avec la base de données pour effectuer des opérations CRUD (Create, Read, Update, Delete). Chaque classe DAO correspond à une entité spécifique du système et fournit des méthodes pour manipuler les données associées dans la base de données.

1. Classe PatientDAO :

* La classe PatientDAO gère les opérations liées aux patients dans la base de données.
* Elle comprend des méthodes telles que insertPatient, selectPatientById, selectAllPatients, updatePatient et deletePatient.
* Ces méthodes permettent d'ajouter, de récupérer, de mettre à jour et de supprimer des enregistrements de patients dans la base de données.

1. Classe DoctorDAO :

* La classe DoctorDAO est responsable de la gestion des opérations liées aux médecins dans la base de données.
* Elle offre des méthodes telles que insertDoctor, selectDoctorById, selectAllDoctors, updateDoctor et deleteDoctor.
* Ces méthodes facilitent l'ajout, la récupération, la mise à jour et la suppression des enregistrements de médecins dans la base de données.

1. Classe AppointmentDAO :

* La classe AppointmentDAO gère les opérations de gestion des rendez-vous dans la base de données.
* Elle inclut des méthodes pour l'ajout, la récupération, la mise à jour et la suppression des rendez-vous, ainsi que la recherche de rendez-vous disponibles, la vérification des conflits d'horaires, etc.

1. Classe ConsultationDAO :

* La classe ConsultationDAO est chargée de la gestion des consultations médicales dans la base de données.
* Elle fournit des méthodes pour ajouter, récupérer, mettre à jour et supprimer des consultations, ainsi que pour rechercher des consultations par patient, par médecin, par date, etc.

1. Classe MedicalRecordDAO :

* La classe MedicalRecordDAO gère les opérations relatives aux dossiers médicaux dans la base de données.
* Elle permet d'ajouter, de récupérer, de mettre à jour et de supprimer des dossiers médicaux, ainsi que de rechercher des dossiers par patient, par médecin, etc.

Chaque classe DAO encapsule la logique d'accès aux données spécifiques à une entité donnée, ce qui permet de séparer clairement les opérations de la logique métier du reste de l'application. Cela favorise une meilleure modularité et facilite la maintenance et l'évolutivité du système.

**5. Classes de Service**

Les classes de service agissent comme des intermédiaires entre l'interface utilisateur graphique (GUI) et les classes DAO (Data Access Object). Elles encapsulent la logique métier de l'application et fournissent une interface pour effectuer des opérations sur les données. Chaque classe de service est dédiée à une entité spécifique du système et fournit des méthodes pour interagir avec les données associées.

1. Classe PatientService:

* La classe PatientService est responsable de la gestion des opérations liées aux patients.
* Elle expose des méthodes telles que addPatient, getAllPatients, getPatientById, updatePatient et deletePatient.
* Ces méthodes permettent d'ajouter, de récupérer, de mettre à jour et de supprimer des enregistrements de patients en utilisant les méthodes correspondantes de la classe PatientDAO.

1. Classe DoctorService:

* La classe DoctorService gère les opérations liées aux médecins.
* Elle offre des méthodes telles que addDoctor, getAllDoctors, getDoctorById, updateDoctor et deleteDoctor.
* Ces méthodes permettent d'ajouter, de récupérer, de mettre à jour et de supprimer des enregistrements de médecins en utilisant les méthodes correspondantes de la classe DoctorDAO.

1. Classe AppointmentService:

* La classe AppointmentService est chargée de la gestion des rendez-vous.
* Elle fournit des méthodes pour ajouter, récupérer, mettre à jour et supprimer des rendez-vous, ainsi que pour gérer les opérations liées aux rendez-vous, telles que la recherche de rendez-vous disponibles et la vérification des conflits d'horaires.
* Ces méthodes utilisent les fonctionnalités exposées par la classe AppointmentDAO pour interagir avec la base de données.

1. Classe ConsultationService:

* La classe ConsultationService gère les opérations liées aux consultations médicales.
* Elle expose des méthodes pour ajouter, récupérer, mettre à jour et supprimer des consultations, ainsi que pour rechercher des consultations par patient, par médecin, par date, etc.
* Ces méthodes utilisent les fonctionnalités fournies par la classe ConsultationDAO pour manipuler les données de consultation dans la base de données.

1. Classe MedicalRecordService:

* La classe MedicalRecordService est responsable de la gestion des dossiers médicaux.
* Elle fournit des méthodes pour ajouter, récupérer, mettre à jour et supprimer des dossiers médicaux, ainsi que pour rechercher des dossiers par patient, par médecin, etc.
* Ces méthodes utilisent les fonctionnalités exposées par la classe MedicalRecordDAO pour interagir avec les données de dossier médical dans la base de données.

Les classes de service jouent un rôle crucial en séparant la logique métier de la logique d'accès aux données, ce qui permet une meilleure modularité et facilite la gestion et l'évolutivité du système.

Elles fournissent une interface claire et cohérente pour effectuer des opérations sur les données, ce qui simplifie le développement de l'interface utilisateur graphique (GUI) et améliore la maintenabilité de l'application.

**6. Panneaux d'Interface Graphique (GUI)**

Les panneaux d'interface graphique (GUI) constituent la partie visible et interactive du système. Chaque panneau est conçu pour gérer une fonctionnalité spécifique du système de gestion de cabinet médical. Voici une description détaillée de chaque panneau GUI :

1. Panneau Patients :

* Ce panneau permet aux utilisateurs d'ajouter, de visualiser, de mettre à jour et de supprimer des informations sur les patients.
* Il comprend des champs de saisie pour le prénom, le nom, le numéro de téléphone et l'adresse e-mail du patient, ainsi que des boutons pour ajouter un patient, afficher tous les patients et supprimer un patient sélectionné.
* Un tableau affiche les détails des patients, avec la possibilité de modifier les informations directement dans le tableau.
* Un menu contextuel permet de supprimer un patient en cliquant avec le bouton droit de la souris sur une ligne du tableau.

1. Panneau Médecins :

* Ce panneau permet aux utilisateurs d'ajouter, de visualiser, de mettre à jour et de supprimer des informations sur les médecins.
* Il comprend des champs de saisie pour le prénom, le nom, le numéro de téléphone, l'adresse e-mail et la spécialité du médecin, ainsi que des boutons pour ajouter un médecin, afficher tous les médecins et supprimer un médecin sélectionné.
* Un tableau affiche les détails des médecins, avec la possibilité de modifier les informations directement dans le tableau.
* Un menu contextuel permet de supprimer un médecin en cliquant avec le bouton droit de la souris sur une ligne du tableau.

1. Panneau Rendez-vous :

* Ce panneau permet aux utilisateurs de planifier et de visualiser les rendez-vous médicaux.
* Il comprend des fonctionnalités pour rechercher des rendez-vous par patient, par médecin et par date, ainsi que pour ajouter, modifier et supprimer des rendez-vous.
* Un calendrier interactif permet de sélectionner la date des rendez-vous, et un tableau affiche les rendez-vous prévus avec des options pour les modifier ou les supprimer.

1. Panneau Consultations :

* Ce panneau permet aux utilisateurs de consulter et de gérer les consultations médicales enregistrées.
* Il offre des fonctionnalités pour rechercher des consultations par patient, par médecin, par date, etc., et pour ajouter, modifier et supprimer des consultations.
* Un tableau affiche les détails des consultations, avec la possibilité de modifier les informations directement dans le tableau.

1. Panneau Dossiers Médicaux :

* Ce panneau permet aux utilisateurs de gérer les dossiers médicaux des patients, y compris les consultations, les ordonnances et les certificats médicaux.
* Il offre des fonctionnalités pour rechercher des dossiers par patient, par médecin, etc., et pour afficher les détails complets des dossiers, y compris les consultations enregistrées et les prescriptions.
* Des boutons permettent d'ajouter de nouvelles consultations, prescriptions et certificats médicaux aux dossiers existants.

Chaque panneau GUI est conçu de manière à offrir une expérience utilisateur intuitive et conviviale, tout en fournissant toutes les fonctionnalités nécessaires pour gérer efficacement les opérations médicales dans un cabinet. L'interface graphique est soigneusement organisée et dispose de contrôles interactifs pour faciliter la navigation et l'interaction avec les données.

**7. Validation des Entrées**

La validation des entrées est une étape cruciale dans tout système informatique pour garantir l'intégrité des données et la fiabilité des opérations. Dans le cadre du système de gestion de cabinet médical, la validation des entrées est effectuée pour chaque saisie utilisateur afin de garantir que les données fournies sont valides et cohérentes. Voici comment la validation des entrées est implémentée dans le système :

1. Validation des Données Patient :

* Lors de l'ajout ou de la mise à jour des informations d'un patient, les champs tels que le prénom, le nom, le numéro de téléphone et l'adresse e-mail sont vérifiés pour s'assurer qu'ils ne sont pas vides et qu'ils respectent les formats appropriés.
* Le prénom et le nom doivent contenir uniquement des lettres.
* Le numéro de téléphone doit être composé uniquement de chiffres.
* L'adresse e-mail doit contenir un symbole "@" et un domaine valide.

1. Validation des Données Médecin :

* De manière similaire aux patients, lors de l'ajout ou de la mise à jour des informations d'un médecin, les champs comme le prénom, le nom, le numéro de téléphone, l'adresse e-mail et la spécialité sont validés pour s'assurer qu'ils sont complets et corrects.
* Les contraintes sur les formats de données sont les mêmes que pour les patients.

1. Validation des Données de Rendez-vous et de Consultation :

* Lors de la planification d'un rendez-vous ou de l'enregistrement d'une consultation, les champs tels que la date, l'heure, le patient, le médecin, le diagnostic et le traitement sont validés.
* Les dates et heures doivent être dans le futur.
* Les champs de patient et de médecin doivent être sélectionnés à partir des listes existantes.
* Le diagnostic et le traitement peuvent être libres, mais la longueur et le format peuvent être vérifiés pour éviter les entrées non valides.

1. Messages d'Erreur et Retours Utilisateur :

* En cas de saisie incorrecte, un message d'erreur approprié est affiché à l'utilisateur pour l'informer de la nature de l'erreur et de la manière de la corriger.
* Les champs incorrects sont souvent mis en évidence visuellement pour attirer l'attention de l'utilisateur sur les zones nécessitant une correction.
* Les messages d'erreur sont formulés de manière conviviale pour aider l'utilisateur à comprendre la nature de l'erreur et à prendre les mesures appropriées pour la corriger.

La validation des entrées garantit que seules des données valides et cohérentes sont acceptées par le système, réduisant ainsi les erreurs et les incohérences dans la base de données. Cela améliore la qualité globale des données et la fiabilité des opérations effectuées dans le système de gestion de cabinet médical.

**8. Relation avec les Objectifs du Projet**

Le système de gestion de cabinet médical a été conçu pour répondre à plusieurs objectifs clés visant à améliorer l'efficacité et la gestion des opérations au sein d'un cabinet médical. Voici comment chaque aspect du projet contribue à la réalisation de ces objectifs :

1. Centralisation des Données :

* Le système centralise toutes les données relatives aux patients, aux médecins, aux rendez-vous, aux consultations et aux dossiers médicaux dans une base de données unique.
* Cela garantit que toutes les informations sont facilement accessibles et mises à jour depuis une seule source, réduisant ainsi les risques d'erreurs et de duplications de données.

1. Automatisation des Tâches Routine :

* L'automatisation des tâches telles que la planification des rendez-vous, l'enregistrement des consultations et la gestion des dossiers médicaux réduit la charge de travail administrative des membres du personnel médical.
* Les fonctionnalités telles que la saisie semi-automatique des données des patients et des médecins accélèrent le processus de saisie et minimisent les erreurs de frappe.

1. Exactitude des Données :

* La validation des entrées garantit que seules des données valides et cohérentes sont acceptées par le système.
* Cela réduit les risques d'erreurs de saisie et de données incorrectes dans la base de données, ce qui améliore la précision et la fiabilité des informations disponibles.

1. Accessibilité Améliorée :

* L'interface utilisateur conviviale du système facilite la navigation et l'utilisation pour tous les membres du personnel du cabinet médical.
* Les fonctionnalités intuitives telles que la recherche de patients et de médecins, ainsi que la visualisation des rendez-vous et des consultations à venir, rendent l'information rapidement accessible à tout moment.

1. Scalabilité :

* Le système est conçu pour être extensible, ce qui permet d'ajouter facilement de nouvelles fonctionnalités et de prendre en charge une croissance future.
* De nouvelles fonctionnalités telles que la gestion des stocks de médicaments, les rapports de performance et l'intégration de systèmes de facturation peuvent être intégrées à mesure que les besoins du cabinet médical évoluent.

Chaque aspect du projet, des classes de base aux interfaces utilisateur en passant par la validation des entrées, contribue à atteindre les objectifs du système de gestion de cabinet médical en améliorant l'efficacité opérationnelle, la précision des données et l'accessibilité des informations.