

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
UNIVERSITÉ DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENE



RAPPORT

JAVA PROJECT

WINDOW BUILDER

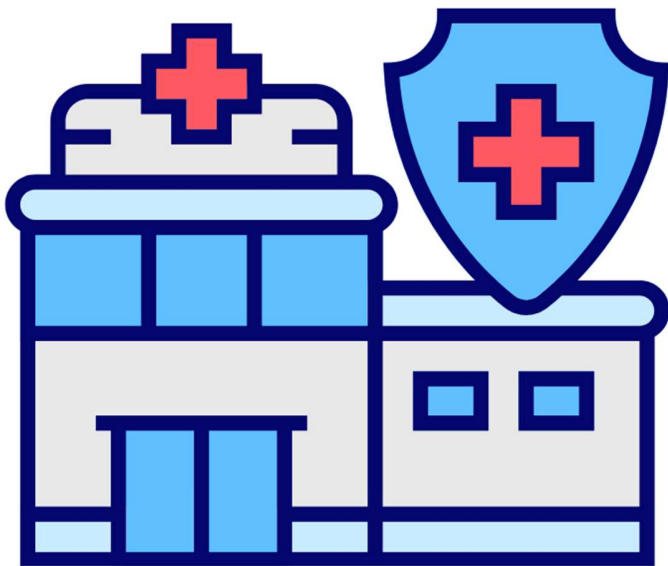
OUALIHINE DOUAA CHARAZED
BOUDJELAL MARIA
DADI SOUMIA
RACHEDI MERIEM

05 / 2024



SOMMAIRE :

1 - Introduction	03
2 - La listes des interfaces	04
3 - Explication du contenue des interfaces	06
4 - Conclusion	



1 - INTRODUCTION :

Ce rapport concerne un projet JAVA qui gère la gestion d'un cabinet médical.

L'objectif principal de ce projet est de développer une sorte d'application informatique en JAVA avec l'outil WINDOW BUILDER de Eclips.

Cet outil permet de créer une interface graphique à notre application ainsi d'avoir une base de données pour stocker tous les données des patients.

Cette application facilite au patients de voir leurs dossier médicale et prendre des rendez-vous et facilite au médecin de gérer ses patients sans confondre les informations et être plus organiser.

Ce rapport expliquera les interfaces créées et leurs fonctionnalités.

2 - LA LISTES DES INTERFACES :

1- Connexion Interface : (Package)

- Accueille.java
- SignIn.java
- SignUp.java

2- Docteur Interface : (Package)

- AfficherRendezVous.java
- AjouterConsultation.java
- ChercherRendezvous.java
- RechercheMatPatient.java
- RemplirDossier.java
- SupprimerDossier.java
- DocteurInterface.java
- RechercheDossier.java

3- Patient Interface : (Package)

- AfficheDossier.java
- AfficheInfo.java
- AfficheRen.java
- AnnulRen.java
- ModifInfo.java
- ModifRen.java
- PrendreRen.java
- PatientPrincipale.java

4- Secrétaire Interface : (Package)

- AfficherRen.java
- AffichPatient.java
- AjoutPatient.java
- AjoutRen.java
- InterfacePrincp.java
- SupprimerRen.java
- SupprimerPatient.java

Connexion interface :

Fichier SignIn :

Généralités :

Le fichier SignIn.java fait partie du module de l'interface de connexion dans le projet de gestion de cabinet médical développé en Java avec Window Builder. Ce fichier est responsable de l'affichage de la fenêtre de connexion où les utilisateurs peuvent saisir leur nom d'utilisateur et leur mot de passe pour accéder au système.

Fonctionnement et utilité :

Ce fichier est crucial car il gère le processus d'authentification des utilisateurs. Lorsque l'application est lancée, cette interface s'affiche. Les utilisateurs doivent fournir leurs identifiants de connexion. Ensuite, le système vérifie ces informations dans la base de données. Si les informations sont correctes, l'utilisateur est redirigé vers l'interface correspondante (par exemple, interface patient, interface docteur, etc.).

Explication des parties importantes :

1. **Connexion à la base de données :**

- Le fichier utilise JDBC pour se connecter à une base de données Oracle. Il charge le pilote Oracle et établit la connexion en utilisant les informations d'identification appropriées (nom d'utilisateur et mot de passe).

2. **Interface graphique :**

- L'interface est construite en utilisant Swing, avec des composants tels que des JLabels, des JTextFields et des JButtons pour saisir les informations de connexion et effectuer la connexion.

3. **Traitement de la connexion :**

- Lorsque l'utilisateur clique sur le bouton "Connexion", un événement est déclenché pour traiter la demande de connexion. Les identifiants saisis par l'utilisateur sont récupérés, puis une requête SQL est exécutée pour vérifier si ces informations correspondent à un compte existant dans la base de données.

- Si les identifiants sont valides, l'utilisateur est connecté et redirigé vers l'interface correspondante. Sinon, un message d'erreur est affiché.

Conclusion :

Le fichier SignIn.java est un composant crucial du système de gestion de cabinet médical, car il gère le processus d'authentification des utilisateurs. En combinant l'interface utilisateur conviviale avec la gestion appropriée des informations d'identification et l'accès à la base de données, ce fichier garantit un accès sécurisé au système pour les utilisateurs autorisés. Son bon fonctionnement est essentiel pour assurer la sécurité et l'efficacité de l'application.

Fichier SignUp :

Généralités :

Le fichier SignUp.java fait partie du module de l'interface de connexion dans le projet de gestion de cabinet médical développé en Java avec Window Builder. Ce fichier est responsable de l'affichage de la fenêtre d'inscription où de nouveaux utilisateurs peuvent créer un compte pour accéder au système.

Fonctionnement et utilité :

Ce fichier est essentiel car il gère le processus d'inscription des nouveaux utilisateurs dans le système. Lorsque l'application est lancée, cette interface s'affiche. Les utilisateurs doivent remplir un formulaire avec des informations personnelles telles que leur nom, prénom, adresse e-mail, etc. Une fois les informations saisies, elles sont enregistrées dans la base de données et un compte utilisateur est créé.

Explication des parties importantes :

1. **Connexion à la base de données :**

- Comme dans le fichier SignIn, SignUp.java utilise JDBC pour se connecter à une base de données Oracle. Il charge le pilote Oracle et établit la connexion en utilisant les informations d'identification appropriées (nom d'utilisateur et mot de passe).

2. **Interface graphique :**

- L'interface est construite en utilisant Swing, avec des composants tels que des JLabels, des JTextFields et des JButtons pour saisir les informations d'inscription et créer un compte.

3. **Traitement de l'inscription :**

- Lorsque l'utilisateur clique sur le bouton "Créer Compte", un événement est déclenché pour traiter la demande d'inscription. Les informations saisies par

l'utilisateur sont récupérées, puis une requête SQL est exécutée pour insérer ces informations dans la base de données.

- Deux requêtes SQL sont exécutées : une pour insérer les informations du nouveau patient dans la table Patient et une autre pour créer un enregistrement dans la table SessionCompte pour gérer l'authentification de l'utilisateur.

Conclusion :

Le fichier SignUp.java est un composant crucial du système de gestion de cabinet médical car il permet aux nouveaux utilisateurs de créer un compte pour accéder au système. En combinant une interface utilisateur intuitive avec un traitement approprié des informations d'inscription et l'accès à la base de données, ce fichier garantit un processus d'inscription fluide et sécurisé pour les utilisateurs. Son bon fonctionnement est essentiel pour permettre aux nouveaux utilisateurs de rejoindre le système et d'en bénéficier pleinement.

Fichier Acceuille :

Généralités :

Le fichier Acceuille.java fait partie du module de l'interface de connexion dans le projet de gestion de cabinet médical développé en Java avec Window Builder. Ce fichier est responsable de l'affichage de la fenêtre d'accueil où les utilisateurs peuvent choisir de se connecter à un compte existant ou de créer un nouveau compte.

Fonctionnement et utilité :

Ce fichier est important car il sert de point d'entrée pour les utilisateurs de l'application. Lorsque l'application est lancée, cette interface s'affiche, offrant aux utilisateurs deux options : se connecter à un compte existant ou créer un nouveau compte. En fonction de leur choix, ils sont redirigés vers l'interface de connexion (SignIn) ou l'interface d'inscription (SignUp).

Explication des parties importantes :

1. *Interface graphique*** :**

- L'interface est construite en utilisant Swing, avec des composants tels que des JLabels et des JButtons pour afficher les options disponibles aux utilisateurs.
- Un fond d'écran est utilisé pour améliorer l'esthétique de l'interface.

2. *Traitement des actions utilisateur*** :**

- Lorsque l'utilisateur clique sur le bouton "Vous Connectez", un événement est déclenché pour ouvrir l'interface de connexion (SignIn).

- Lorsque l'utilisateur clique sur le bouton "Créez Un Compte", un événement est déclenché pour ouvrir l'interface d'inscription (SignUp).

Conclusion :

Le fichier Acceuille.java joue un rôle crucial dans le processus d'interaction entre l'utilisateur et l'application. En fournissant des options claires pour se connecter ou créer un compte, il facilite la navigation des utilisateurs dans l'application. Son interface conviviale et ses fonctionnalités de redirection appropriées contribuent à une expérience utilisateur agréable et intuitive. Son bon fonctionnement est essentiel pour garantir que les utilisateurs peuvent accéder facilement aux fonctionnalités de l'application et utiliser le système de gestion de cabinet médical de manière efficace.

Patient Interface :

Fichier AfficheDossier :

Généralités :

Le fichier AfficheDossier.java fait partie du module de l'interface patient dans le projet de gestion de cabinet médical développé en Java avec Window Builder. Ce fichier est responsable de l'affichage des dossiers médicaux d'un patient, y compris les consultations passées et les détails sur le patient lui-même.

Fonctionnement et utilité :

Ce fichier est crucial car il permet aux médecins et au personnel médical d'accéder rapidement aux informations médicales importantes d'un patient. Lorsque l'application est lancée, cette interface est utilisée pour afficher les détails du patient, y compris son nom, son prénom, son groupe sanguin et ses consultations passées. Cela permet aux médecins de consulter efficacement l'historique médical du patient et de prendre des décisions éclairées concernant son traitement.

Explication des parties importantes :

1. **Connexion à la base de données :**

- Tout comme dans les autres fichiers, AfficheDossier utilise JDBC pour se connecter à une base de données Oracle. Il charge le pilote Oracle et établit la connexion en utilisant les informations d'identification appropriées (nom d'utilisateur et mot de passe).

2. **Interface graphique :**

- L'interface est construite en utilisant Swing, avec des composants tels que des JLabels et une JTable pour afficher les informations sur le patient et ses consultations.
- Un fond d'écran est utilisé pour améliorer l'esthétique de l'interface.

3. **Affichage des données du patient :**

- Les détails du patient, tels que le nom, le prénom et le groupe sanguin, sont extraits de la base de données et affichés dans les JLabels appropriés.
- Les consultations passées du patient sont récupérées de la base de données et affichées dans une JTable, rendant ainsi les informations facilement consultables et organisées.

Conclusion :

Le fichier AfficheDossier.java est un composant essentiel de l'interface patient dans le système de gestion de cabinet médical. En fournissant un accès rapide et organisé aux informations médicales d'un patient, il permet aux médecins et au personnel médical de prendre des décisions éclairées concernant le traitement des patients. Son bon fonctionnement et son interface conviviale sont essentiels pour assurer une expérience utilisateur optimale et garantir une gestion efficace des dossiers médicaux dans le cabinet médical.

Fichier AfficheInfo :

Généralités :

Le fichier AfficheInfo.java fait partie du module de l'interface patient dans le projet de gestion de cabinet médical développé en Java avec Window Builder. Ce fichier est responsable de l'affichage des informations détaillées sur un patient, telles que son nom, prénom, adresse, maladies, etc.

Fonctionnement et utilité :

Ce fichier est crucial car il permet aux médecins et au personnel médical d'accéder rapidement aux informations personnelles d'un patient. Lorsque l'application est lancée, cette interface est utilisée pour afficher les détails du patient associé à un matricule spécifique. Cela permet aux médecins de consulter efficacement les détails du patient et de prendre des décisions éclairées concernant son traitement.

Explication des parties importantes :

1. **Connexion à la base de données :**

- Tout comme dans les autres fichiers, AfficheInfo utilise JDBC pour se connecter à une base de données Oracle. Il charge le pilote Oracle et établit la connexion en utilisant les informations d'identification appropriées (nom d'utilisateur et mot de passe).

2. **Interface graphique :**

- L'interface est construite en utilisant Swing, avec des composants tels que des JLabels pour afficher les informations sur le patient.
- Un fond d'écran est utilisé pour améliorer l'esthétique de l'interface.

3. **Affichage des informations du patient :**

- Les détails du patient, tels que le nom, le prénom, l'email, le genre, les maladies, etc., sont extraits de la base de données et affichés dans les JLabels appropriés.
- Les informations sont récupérées en fonction du matricule du patient fourni en paramètre lors de l'initialisation de la classe.

Conclusion :

Le fichier AfficheInfo.java est un composant essentiel de l'interface patient dans le système de gestion de cabinet médical. En fournissant un accès rapide et organisé aux informations personnelles d'un patient, il permet aux médecins et au personnel médical de prendre des décisions éclairées concernant le traitement des patients. Son bon fonctionnement et son interface conviviale sont essentiels pour assurer une expérience utilisateur optimale et garantir une gestion efficace des informations des patients dans le cabinet médical.

Fichier AfficheRen :

Généralités :

Le fichier AfficheRen.java fait partie du module de l'interface patient dans le projet de gestion de cabinet médical développé en Java avec Window Builder. Ce fichier est responsable de l'affichage des rendez-vous d'un patient spécifique.

Fonctionnement et utilité :

Ce fichier est essentiel pour permettre aux médecins et au personnel médical de consulter les rendez-vous prévus pour un patient donné. Lorsque l'application est

lancée, cette interface est utilisée pour afficher tous les rendez-vous associés à un matricule de patient spécifique. Cela permet aux médecins de planifier leurs consultations et de gérer efficacement leur emploi du temps.

Explication des parties importantes :

1. **Connexion à la base de données :**

- Tout comme dans les autres fichiers, AfficheRen utilise JDBC pour se connecter à une base de données Oracle. Il charge le pilote Oracle et établit la connexion en utilisant les informations d'identification appropriées (nom d'utilisateur et mot de passe).

2. **Interface graphique :**

- L'interface est minimaliste, avec seulement une JTable pour afficher les rendez-vous du patient.
- Un fond d'écran est utilisé pour améliorer l'esthétique de l'interface.

3. **Affichage des rendez-vous :**

- Les rendez-vous du patient sont extraits de la base de données en fonction de son matricule.
- La requête SQL sélectionne tous les rendez-vous associés au matricule du patient spécifié.
- Les données récupérées sont ensuite passées à un modèle de table pour affichage dans la JTable.
- Si le nombre de rendez-vous est important, la JTable devient scrollable pour une meilleure expérience utilisateur.

Conclusion :

Le fichier AfficheRen.java est un composant crucial de l'interface patient dans le système de gestion de cabinet médical. En fournissant un accès rapide et organisé aux rendez-vous planifiés pour un patient spécifique, il permet aux médecins de planifier efficacement leurs consultations et de garantir une gestion optimale de leur emploi du temps. Son bon fonctionnement et son interface conviviale sont essentiels pour assurer une expérience utilisateur optimale et garantir une gestion efficace des rendez-vous dans le cabinet médical.

Fichier AnnulRen :

Généralités :

Le fichier AnnulRen.java fait partie du module de l'interface patient dans le projet de gestion de cabinet médical développé en Java avec Window Builder. Ce fichier

est responsable de l'interface permettant d'annuler un rendez-vous pour un patient spécifique.

Fonctionnement et utilité :

Ce fichier est essentiel pour permettre aux patients d'annuler un rendez-vous qu'ils ont précédemment fixé avec leur médecin. Lorsque l'application est lancée, cette interface est utilisée pour saisir la date et l'heure du rendez-vous à annuler. Une fois que les informations sont fournies, l'utilisateur peut cliquer sur le bouton "Annuler" pour supprimer le rendez-vous de la base de données.

Explication des parties importantes :

1. **Connexion à la base de données :**

- Tout comme dans les autres fichiers, AnnulRen utilise JDBC pour se connecter à une base de données Oracle. Il charge le pilote Oracle et établit la connexion en utilisant les informations d'identification appropriées (nom d'utilisateur et mot de passe).

2. **Interface graphique :**

- L'interface permet à l'utilisateur de saisir la date et l'heure du rendez-vous à annuler.
- Un bouton "Annuler" permet à l'utilisateur de soumettre les informations pour annuler le rendez-vous.
- Un fond d'écran est utilisé pour améliorer l'esthétique de l'interface.

3. **Annulation du rendez-vous :**

- Lorsque l'utilisateur clique sur le bouton "Annuler", les informations saisies sont récupérées.
- Une requête SQL DELETE est construite pour supprimer le rendez-vous correspondant à la date et l'heure spécifiées.
- La requête est exécutée et le rendez-vous est supprimé de la base de données.
- Un message de confirmation est affiché pour informer l'utilisateur que le rendez-vous a été annulé avec succès.

Conclusion :

Le fichier AnnulRen.java est un composant important de l'interface patient dans le système de gestion de cabinet médical. Il offre aux patients une méthode conviviale pour annuler les rendez-vous prévus avec leur médecin. Son bon fonctionnement et son interface intuitive sont essentiels pour assurer une expérience utilisateur optimale et garantir une gestion efficace des rendez-vous dans le cabinet médical.

Fichier ModifInfo :

Généralités :

Le fichier ModifInfo.java fait partie du module de l'interface patient dans le projet de gestion de cabinet médical développé en Java avec Window Builder. Ce fichier est responsable de l'interface permettant de modifier les informations personnelles d'un patient.

Fonctionnement et utilité :

Ce fichier est essentiel pour permettre aux patients de mettre à jour leurs informations personnelles, telles que leur nom, prénom, adresse e-mail, numéro de téléphone, etc. Lorsque l'application est lancée, cette interface est utilisée pour saisir les nouvelles informations et les soumettre pour mise à jour.

Explication des parties importantes :

1. **Connexion à la base de données :**

- Tout comme dans les autres fichiers, ModifInfo utilise JDBC pour se connecter à une base de données Oracle. Il charge le pilote Oracle et établit la connexion en utilisant les informations d'identification appropriées (nom d'utilisateur et mot de passe).

2. **Interface graphique :**

- L'interface permet à l'utilisateur de saisir les nouvelles informations personnelles dans des champs de texte et de sélectionner le genre à l'aide de boutons radio.
- Un bouton "Modifier" permet à l'utilisateur de soumettre les nouvelles informations pour mise à jour.

3. **Mise à jour des informations :**

- Lorsque l'utilisateur clique sur le bouton "Modifier", les nouvelles informations saisies sont récupérées.
- Une requête SQL UPDATE est construite pour mettre à jour les informations du patient dans la base de données.
- La requête est exécutée et les informations du patient sont mises à jour.
- Un message de confirmation est affiché pour informer l'utilisateur que les informations ont été mises à jour avec succès.

Conclusion :

Le fichier ModifInfo.java est un composant important de l'interface patient dans le système de gestion de cabinet médical. Il offre aux patients une méthode conviviale pour mettre à jour leurs informations personnelles. Son bon fonctionnement et son interface intuitive sont essentiels pour assurer une expérience utilisateur optimale et garantir une gestion efficace des informations des patients dans le cabinet médical.

Fichier ModifRen :

Généralités :

Le fichier ModifRen.java est un composant de l'interface patient dans le projet de gestion de cabinet médical développé en Java avec Window Builder. Il permet de modifier les rendez-vous existants dans le système.

Fonctionnement et utilité :

Ce fichier offre aux utilisateurs la possibilité de modifier les détails des rendez-vous en fournissant les anciennes et les nouvelles informations. L'utilisateur peut choisir entre deux types de médecins : le médecin principal et le médecin remplaçant. Lorsque l'utilisateur modifie un rendez-vous, la base de données est mise à jour en conséquence.

Explication des parties importantes :

1. *Connexion à la base de données*** :**

- Tout comme dans les autres fichiers, ModifRen utilise JDBC pour se connecter à une base de données Oracle. Il charge le pilote Oracle et établit la connexion en utilisant les informations d'identification appropriées (nom d'utilisateur et mot de passe).

2. *Interface graphique*** :**

- L'interface permet à l'utilisateur de saisir les anciennes et les nouvelles dates ainsi que les anciennes et les nouvelles heures pour le rendez-vous.
- L'utilisateur peut choisir entre deux types de médecins à l'aide de boutons.
- Deux boutons "Modifier" sont fournis, un pour chaque type de médecin.

3. *Mise à jour des rendez-vous*** :**

- Lorsque l'utilisateur clique sur le bouton "Modifier" correspondant au type de médecin choisi, les informations saisies sont récupérées.

- Une requête SQL UPDATE est construite pour mettre à jour le rendez-vous dans la base de données en fonction des informations fournies.
- La requête est exécutée et le rendez-vous est mis à jour dans la base de données.
- Un message de confirmation est affiché pour informer l'utilisateur que le rendez-vous a été modifié avec succès.

Conclusion :

Le fichier ModifRen.java est un composant important de l'interface patient dans le système de gestion de cabinet médical. Il offre aux utilisateurs une méthode conviviale pour modifier les détails de leurs rendez-vous. Son bon fonctionnement et son interface intuitive sont essentiels pour assurer une gestion efficace des rendez-vous dans le cabinet médical.

Fichier PrendreRen :

Généralités :

Le fichier PrendreRen.java fait partie de l'interface patient dans le projet de gestion de cabinet médical développé en Java avec Window Builder. Son rôle est de permettre aux patients de prendre de nouveaux rendez-vous avec les médecins du cabinet.

Fonctionnement et utilité :

Ce fichier offre aux patients la possibilité de choisir entre le médecin principal et le médecin remplaçant pour prendre un rendez-vous. L'utilisateur doit fournir la date et l'heure souhaitées pour le rendez-vous. Une fois les informations saisies, elles sont enregistrées dans la base de données du cabinet médical.

Explication des parties importantes :

1. *Connexion à la base de données*** :**

- Comme dans les autres fichiers, PrendreRen utilise JDBC pour se connecter à une base de données Oracle. Il charge le pilote Oracle et établit la connexion en utilisant les informations d'identification appropriées (nom d'utilisateur et mot de passe).

2. *Interface graphique*** :**

- L'interface permet à l'utilisateur de saisir la date et l'heure du rendez-vous.
- Deux boutons sont fournis pour permettre au patient de choisir entre le médecin principal et le médecin remplaçant.

3. **Enregistrement du rendez-vous :**

- Lorsque l'utilisateur clique sur l'un des boutons pour prendre rendez-vous, les informations saisies sont récupérées.
- Une requête SQL INSERT est construite pour enregistrer le nouveau rendez-vous dans la base de données en fonction des informations fournies.
- La requête est exécutée et le rendez-vous est ajouté à la base de données.
- Un message de confirmation est affiché pour informer l'utilisateur que le rendez-vous a été pris avec succès.

Conclusion :

Le fichier PrendreRen.java joue un rôle crucial dans l'interface patient du système de gestion de cabinet médical. Il offre aux patients une méthode conviviale pour prendre de nouveaux rendez-vous avec les médecins du cabinet. Son bon fonctionnement et son interface intuitive sont essentiels pour assurer une bonne gestion des rendez-vous et une expérience utilisateur optimale.

Fichier PatientPrincipale :

Généralités :

Le fichier PatientPrincipale.java constitue l'une des interfaces principales destinées aux patients dans le projet de gestion de cabinet médical développé en Java. Son objectif est de fournir aux patients un ensemble d'options pour gérer leurs rendez-vous et leurs informations personnelles.

Fonctionnement et utilité :

Ce fichier offre aux patients un menu principal qui leur permet d'accéder à différentes fonctionnalités de l'interface patient. Chaque option du menu déclenche l'ouverture d'une nouvelle fenêtre dédiée à une tâche spécifique, comme afficher les informations personnelles du patient, modifier ces informations, prendre un rendez-vous, etc.

Explication des parties importantes :

1. **Connexion à d'autres fonctionnalités :**

- Le menu principal comprend plusieurs boutons, chacun étant associé à une fonctionnalité spécifique.
- Lorsque l'utilisateur clique sur l'un de ces boutons, une nouvelle fenêtre correspondant à la fonctionnalité sélectionnée est ouverte.

2. **Interface graphique :**

- L'interface est construite avec Swing et utilise un ensemble de boutons pour représenter les différentes fonctionnalités disponibles.
- Un fond d'écran est chargé à partir d'une image PNG, ce qui améliore l'esthétique de l'interface.

3. **Gestion des actions utilisateur :**

- Chaque bouton est associé à un gestionnaire d'événements de clic de souris qui détermine l'action à entreprendre lorsque l'utilisateur clique sur le bouton.
- Par exemple, cliquer sur le bouton "Afficher Informations" ouvre une fenêtre pour afficher les informations personnelles du patient.

Conclusion :

Le fichier PatientPrincipale.java joue un rôle central dans l'interface patient du système de gestion de cabinet médical. Il fournit un accès facile aux différentes fonctionnalités disponibles pour les patients, ce qui améliore l'expérience utilisateur et facilite la gestion de leurs rendez-vous et de leurs informations médicales. Son interface conviviale et ses fonctionnalités clairement présentées contribuent à une utilisation intuitive de l'application par les patients.

Docteur Interface :

Fichier AfficheRendezvous :

Généralités :

Ce fichier Java appartient au dossier `docteurInterface` et est nommé `AfficheRendezvous.java`. Il s'agit d'une classe qui affiche les rendez-vous d'un médecin dans une interface graphique. Le code utilise des composants Swing pour créer une fenêtre JFrame contenant une JTable pour afficher les rendez-vous récupérés à partir d'une base de données Oracle.

Fonctionnement et Utilité :

Le fichier permet à un médecin de visualiser ses rendez-vous à travers une interface utilisateur conviviale. Lorsque l'application est lancée, elle établit une connexion à une base de données Oracle contenant des informations sur les rendez-vous. Ensuite, elle exécute une requête SQL pour sélectionner les rendez-vous du médecin en fonction de son statut. Les résultats de la requête sont ensuite affichés dans une JTable dans la fenêtre graphique de l'application.

Explication des Parties Importantes :

- 1. ***Initialisation de la Fenêtre et des Composants Swing*** : Le** code initialise une fenêtre JFrame et une JTable pour afficher les données.
- 2. ***Connexion à la Base de Données Oracle*** :** La connexion à la base de données Oracle est établie en chargeant le pilote JDBC et en spécifiant les informations de connexion telles que l'URL, le nom d'utilisateur et le mot de passe.
- 3. ***Requête SQL pour Récupérer les Rendez-vous du Médecin*** :** Une requête SQL est définie pour sélectionner les rendez-vous du médecin en fonction de son statut.
- 4. ***Exécution de la Requête et Affichage des Résultats*** :** La requête SQL est exécutée à l'aide d'un objet Statement, et les résultats sont récupérés dans un objet ResultSet. Ensuite, la méthode `resultSetToTableModel` de la bibliothèque DbUtils est utilisée pour convertir le ResultSet en un modèle de table pouvant être affiché dans une JTable.
- 5. ***Affichage dans la JTable*** :** Les données récupérées sont affichées dans une JTable, et un JScrollPane est utilisé pour rendre la table scrollable si nécessaire.

Conclusion :

Le fichier `AfficheRendezvous.java` offre une fonctionnalité importante dans le système de gestion de cabinet médical. Il permet aux médecins de visualiser rapidement et facilement leurs rendez-vous, ce qui peut contribuer à une meilleure gestion du temps et des ressources. En utilisant des composants Swing et JDBC, il fournit une interface utilisateur intuitive pour interagir avec les données de la base de données Oracle.

Fichier AjouterConsultation :

Généralités :

Ce fichier Java appartient au dossier `docteurInterface` et est nommé `AjouterConsultation.java`. Il s'agit d'une classe qui permet aux médecins d'ajouter de nouvelles consultations pour les patients dans une interface graphique. Le code utilise des composants Swing pour créer une fenêtre JFrame avec des champs de saisie pour le matricule du patient, le diagnostic, le médicament prescrit et la date de la consultation.

Fonctionnement et Utilité :

Le fichier offre une fonctionnalité essentielle dans le système de gestion de cabinet médical en permettant aux médecins d'enregistrer de nouvelles consultations pour leurs patients. Lorsque l'application est lancée, les médecins peuvent saisir les détails de la consultation, y compris le matricule du patient, le diagnostic, le médicament prescrit et la date de la consultation. En cliquant sur le bouton "Confirmer", les informations saisies sont insérées dans la base de données Oracle, permettant ainsi de garder une trace précise des consultations médicales.

Explication des Parties Importantes :

1. **Initialisation de la Fenêtre et des Composants Swing : Le** code initialise une fenêtre JFrame et des champs de texte pour saisir les informations de la consultation.

2. **Connexion à la Base de Données Oracle :** La connexion à la base de données Oracle est établie en chargeant le pilote JDBC et en spécifiant les informations de connexion telles que l'URL, le nom d'utilisateur et le mot de passe.

3. **Saisie des Informations de Consultation :** Les médecins saisissent le matricule du patient, le diagnostic, le médicament prescrit et la date de la consultation dans les champs de texte prévus à cet effet.

4. **Ajout de la Consultation dans la Base de Données :** Lorsque le bouton "Confirmer" est cliqué, les informations saisies sont récupérées et utilisées pour construire une instruction SQL d'insertion. Cette instruction est ensuite exécutée à l'aide d'un objet Statement, ce qui permet d'ajouter la nouvelle consultation à la base de données.

5. **Gestion des Erreurs :** En cas d'erreur lors de l'ajout de la consultation, un message d'erreur est affiché pour informer l'utilisateur de l'échec de l'opération.

Conclusion :

Le fichier `AjouterConsultation.java` fournit une fonctionnalité cruciale dans le système de gestion de cabinet médical en permettant aux médecins d'ajouter de nouvelles consultations pour les patients. En utilisant des composants Swing et JDBC, il offre une interface utilisateur intuitive pour saisir et enregistrer les informations de consultation dans la base de données Oracle.

Fichier ChercherRendezvous :

Généralités :

Ce fichier Java appartient au dossier `docteurInterface` et est nommé `ChercherRendezvous.java`. Il s'agit d'une classe qui permet aux médecins de rechercher les rendez-vous pour une date spécifique dans une interface graphique. Le code utilise des composants Swing pour créer une fenêtre JFrame avec un champ de saisie pour la date et un bouton pour confirmer la recherche.

Fonctionnement et Utilité :

Ce fichier offre une fonctionnalité importante dans le système de gestion de cabinet médical en permettant aux médecins de consulter les rendez-vous pour une date particulière. Lorsque l'application est lancée, les médecins saisissent une date dans le champ prévu à cet effet, puis cliquent sur le bouton "Confirmer" pour afficher les rendez-vous correspondants dans un tableau. Cela leur permet de mieux organiser leur emploi du temps et de préparer leurs consultations.

Explication des Parties Importantes :

1. **Initialisation de la Fenêtre et des Composants Swing :** Le code initialise une fenêtre JFrame et des composants Swing tels qu'un champ de texte pour la saisie de la date et un bouton pour confirmer la recherche.

2. **Connexion à la Base de Données Oracle :** La connexion à la base de données Oracle est établie en chargeant le pilote JDBC et en spécifiant les informations de connexion telles que l'URL, le nom d'utilisateur et le mot de passe.

3. **Recherche des Rendez-vous :** Lorsque le bouton "Confirmer" est cliqué, le code récupère la date saisie par l'utilisateur, puis construit une requête SQL pour sélectionner les rendez-vous pour cette date dans la base de données. La requête est exécutée à l'aide d'un objet Statement, et les résultats sont affichés dans un tableau.

4. **Gestion des Erreurs :** En cas d'erreur lors de la recherche des rendez-vous, un message d'erreur est affiché pour informer l'utilisateur de l'échec de l'opération.

Conclusion :

Le fichier `ChercherRendezvous.java` offre une fonctionnalité pratique pour les médecins dans le système de gestion de cabinet médical en leur permettant de rechercher les rendez-vous pour une date spécifique. En utilisant des composants Swing et JDBC, il fournit une interface utilisateur conviviale pour saisir la date et afficher les rendez-vous correspondants. Cela contribue à une meilleure gestion

de l'emploi du temps des médecins et à une organisation efficace des consultations.

Fichier RechercheDossier

Généralités :

Ce fichier Java, nommé `RechercheDossier.java`, appartient au dossier `docteurInterface`. Il s'agit d'une classe qui permet aux médecins de rechercher et d'afficher les dossiers des patients en saisissant leur matricule dans une interface graphique. Le code utilise des composants Swing pour créer une fenêtre JFrame avec un champ de saisie pour le matricule du patient et un bouton pour confirmer la recherche.

Fonctionnement et Utilité :

Ce fichier offre une fonctionnalité importante dans le système de gestion de cabinet médical en permettant aux médecins de rechercher et de visualiser les dossiers des patients. Lorsque l'application est lancée, les médecins saisissent le matricule du patient dans le champ prévu à cet effet, puis cliquent sur le bouton "Confirmer" pour afficher le dossier du patient correspondant dans une nouvelle fenêtre.

Explication des Parties Importantes :

1. **Initialisation de la Fenêtre et des Composants Swing :** Le code initialise une fenêtre JFrame et des composants Swing tels qu'un champ de texte pour la saisie du matricule du patient et un bouton pour confirmer la recherche.

2. **Affichage du Dossier du Patient :** Lorsque le bouton "Confirmer" est cliqué, le code crée une instance de la classe `AfficheDossier` (qui appartient au dossier `patientInterface`) en lui transmettant le matricule du patient saisi. Cette classe est responsable de l'affichage du dossier du patient dans une nouvelle fenêtre.

3. **Gestion des Erreurs :** En cas d'erreur lors de l'affichage du dossier du patient, des mécanismes de gestion des erreurs pourraient être ajoutés pour informer l'utilisateur de l'échec de l'opération.

Conclusion :

Le fichier `RechercheDossier.java` offre une fonctionnalité essentielle pour les médecins dans le système de gestion de cabinet médical en leur permettant de rechercher et de visualiser les dossiers des patients. En utilisant des composants Swing, il fournit une interface utilisateur conviviale pour saisir le matricule du

patient et afficher le dossier correspondant. Cela contribue à une meilleure gestion des informations des patients et à une prestation de soins plus efficace.

Fichier RechercheMatPatient

Généralités :

Ce fichier Java, nommé `RechercheMatPatient.java`, fait partie du dossier `docteurInterface`. Il s'agit d'une classe qui permet aux médecins de rechercher des patients par leur nom dans une interface graphique. Le code utilise des composants Swing pour créer une fenêtre JFrame avec un champ de saisie pour le nom du patient et un bouton pour confirmer la recherche.

Fonctionnement et Utilité :

Ce fichier offre une fonctionnalité importante dans le système de gestion de cabinet médical en permettant aux médecins de rechercher des patients par leur nom. Lorsque l'application est lancée, les médecins saisissent le nom du patient dans le champ prévu à cet effet, puis cliquent sur le bouton "Confirmer" pour afficher les résultats de la recherche dans un tableau.

Explication des Parties Importantes :

1. *Initialisation de la Fenêtre et des Composants Swing***** : Le code initialise une fenêtre JFrame et des composants Swing tels qu'un champ de texte pour la saisie du nom du patient et un bouton pour confirmer la recherche.

2. *Affichage des Résultats de Recherche***** : Lorsque le bouton "Confirmer" est cliqué, le code exécute une requête SQL pour rechercher les patients dont le nom correspond à celui saisi par l'utilisateur. Les résultats de la recherche sont ensuite affichés dans un tableau grâce à l'utilisation de la bibliothèque `DbUtils`.

3. *Gestion des Erreurs***** : En cas d'erreur lors de la recherche des patients, des mécanismes de gestion des erreurs sont mis en place pour informer l'utilisateur de l'échec de l'opération.

Conclusion :

Le fichier `RechercheMatPatient.java` offre une fonctionnalité essentielle pour les médecins dans le système de gestion de cabinet médical en leur permettant de rechercher des patients par leur nom. En utilisant des composants Swing, il fournit une interface utilisateur conviviale pour saisir le nom du patient et afficher les résultats de la recherche dans un tableau. Cela contribue à une meilleure gestion des informations des patients et à une prestation de soins plus efficace.

Fichier RemplirDossier

Généralités :

Ce fichier Java, nommé `RemplirDossier.java`, fait partie du dossier `docteurInterface`. Il s'agit d'une classe qui permet aux médecins de remplir les dossiers médicaux des patients, y compris les informations sur les consultations. Le code utilise des composants Swing pour créer une fenêtre JFrame avec des champs de saisie pour les détails du patient et de la consultation.

Fonctionnement et Utilité :

Ce fichier offre une fonctionnalité essentielle dans le système de gestion de cabinet médical en permettant aux médecins de créer de nouveaux dossiers médicaux pour les patients et d'ajouter des informations sur leurs consultations. Lorsque l'application est lancée, les médecins saisissent les détails du patient, tels que le matricule et le groupe sanguin, ainsi que les détails de la consultation, tels que le diagnostic, les médicaments prescrits et la date. En cliquant sur le bouton "Confirmer", ces informations sont insérées dans les tables correspondantes de la base de données.

Explication des Parties Importantes :

1. **Initialisation de la Fenêtre et des Composants Swing** : Le code initialise une fenêtre JFrame et des composants Swing tels que des champs de texte pour la saisie des détails du patient et de la consultation, ainsi qu'un bouton pour confirmer l'ajout.

2. **Ajout du Dossier Médical** : Lorsque le bouton "Confirmer" est cliqué, le code exécute une requête SQL pour insérer le dossier médical du patient dans la table correspondante de la base de données.

3. **Ajout de la Consultation** : Ensuite, le code exécute une autre requête SQL pour ajouter les détails de la consultation dans la table correspondante de la base de données.

4. **Gestion des Erreurs** : Des mécanismes de gestion des erreurs sont mis en place pour informer l'utilisateur en cas d'échec de l'opération d'ajout.

Conclusion :

Le fichier `RemplirDossier.java` offre une fonctionnalité cruciale dans le système de gestion de cabinet médical en permettant aux médecins de créer de nouveaux dossiers médicaux pour les patients et de documenter leurs consultations. En utilisant des composants Swing, il fournit une interface utilisateur intuitive pour saisir les informations pertinentes et les enregistrer dans la base de données,

contribuant ainsi à une meilleure gestion des données des patients et à une prestation de soins plus efficace.

Fichier SupprimerDossier :

Généralités :

Ce fichier Java, nommé `SupprimerDossier.java`, fait partie du dossier `docteurInterface`. Il s'agit d'une classe qui permet aux médecins de supprimer les dossiers médicaux des patients, ainsi que toutes les consultations associées. Le code utilise des composants Swing pour créer une fenêtre JFrame avec un champ de saisie pour le matricule du patient et un bouton pour confirmer la suppression.

Fonctionnement et Utilité :

Ce fichier offre une fonctionnalité importante dans le système de gestion de cabinet médical en permettant aux médecins de supprimer les dossiers médicaux des patients. Lorsque l'application est lancée, les médecins saisissent le matricule du patient dont ils veulent supprimer le dossier. En cliquant sur le bouton "Confirmer", toutes les consultations associées à ce patient sont d'abord supprimées de la base de données, puis le dossier médical du patient est supprimé.

Explication des Parties Importantes :

1. **Initialisation de la Fenêtre et des Composants Swing :** Le code initialise une fenêtre JFrame et des composants Swing tels qu'un champ de texte pour la saisie du matricule du patient et un bouton pour confirmer la suppression.

2. **Suppression des Consultations et du Dossier Médical :** Lorsque le bouton "Confirmer" est cliqué, le code exécute des requêtes SQL pour supprimer toutes les consultations et le dossier médical du patient de la base de données.

3. **Gestion des Erreurs :** Des mécanismes de gestion des erreurs sont mis en place pour informer l'utilisateur en cas d'échec de l'opération de suppression.

Conclusion :

Le fichier `SupprimerDossier.java` offre une fonctionnalité essentielle dans le système de gestion de cabinet médical en permettant aux médecins de supprimer les dossiers médicaux des patients. En utilisant des composants Swing, il fournit une interface utilisateur simple pour saisir le matricule du patient à supprimer et confirmer l'opération de suppression. Cette fonctionnalité contribue à maintenir la base de données du cabinet médical propre et à jour en permettant la gestion efficace des dossiers des patients.

Fichier DocteurInterface :

Généralités :

Ce fichier Java, nommé `DocteurInterface.java`, fait partie du dossier `docteurInterface`. Il s'agit de l'interface principale destinée aux médecins dans le système de gestion de cabinet médical. L'interface fournit des boutons pour accéder à différentes fonctionnalités telles que remplir un dossier, chercher un dossier, supprimer un dossier, afficher les rendez-vous, rechercher un patient par son matricule, chercher un rendez-vous et ajouter une consultation.

Fonctionnement et Utilité :

Ce fichier offre une interface utilisateur conviviale permettant aux médecins d'accéder rapidement aux fonctionnalités essentielles du système de gestion de cabinet médical. Les boutons permettent d'accéder à différentes fonctionnalités, comme ajouter, chercher ou supprimer des dossiers, afficher les rendez-vous, rechercher des patients ou des rendez-vous, et ajouter des consultations. L'utilisation de l'interface facilite la navigation et améliore l'efficacité des médecins dans la gestion des dossiers et des rendez-vous.

Explication des Parties Importantes :

1. **Initialisation de la Fenêtre et des Composants Swing :** Le code initialise une fenêtre JFrame et des boutons Swing pour chaque fonctionnalité.

2. **Événements des Boutons :** Chaque bouton est associé à un événement de clic de souris qui déclenche l'ouverture de la fenêtre correspondante pour la fonctionnalité sélectionnée.

3. **Gestion de la Déconnexion :** Un bouton "Déconnexion" est fourni pour permettre aux médecins de se déconnecter de l'interface principale.

Conclusion :

Le fichier `DocteurInterface.java` fournit une interface utilisateur complète pour les médecins dans le système de gestion de cabinet médical. En regroupant toutes les fonctionnalités essentielles sous forme de boutons dans une seule interface, il offre une expérience utilisateur conviviale et efficace. Cette interface facilite la gestion des dossiers médicaux et des rendez-vous, ce qui contribue à améliorer l'efficacité et la productivité des médecins dans leur pratique quotidienne.

Secrétaire Interface :

Fichier Ajoutpatient :

Généralités :

Le fichier Ajoutpatient.java fait partie de l'interface secrétaire dans le projet de gestion de cabinet médical développé en Java avec Window Builder. Son rôle est d'ajouter un nouveau patient à une base de données Oracle via un formulaire graphique

Fonctionnement et utilité :

Cette interface permet au secrétaire du cabinet médical d'ajouter de nouveaux patients à la base de données. La secrétaire doit fournir les informations suivantes : nom, prénom, date de naissance, historique médicale, adresse, téléphone, email et genre. Une fois les informations saisies, elles sont enregistrées dans la base de données du cabinet médical

Explication des parties importantes :

Connexion à la base de données:

Utilise JDBC pour se connecter à une base de données Oracle, charge le pilote Oracle JDBC, et établit la connexion avec les informations d'identification appropriées

Interface graphique:

Permet à la secrétaire de saisir les informations du patient : nom, prénom, date de naissance, historique médicale, adresse, téléphone, email et genre

Fournit des champs de texte pour la saisie et un bouton "Confirmer" pour soumettre les informations

Enregistrement du patient:

Lors du clic sur "Confirmer", les informations saisies sont récupérées et une requête SQL **INSERT** est construite pour ajouter le patient dans la base de données

La requête est exécutée, validant la transaction en cas de succès ou annulant en cas d'erreur

Un message de confirmation ou d'erreur est affiché à la secrétaire

Cette interface facilite l'ajout de nouveaux patients dans la base de données, automatisant et sécurisant le processus de saisie des informations

Fichier SupprimPatient :

Généralités :

Le fichier SupprimPatient.java fait partie du module de l'interface secrétaire dans le projet de gestion de cabinet médical développé en Java avec Window Builder. Ce fichier est responsable de l'interface permettant de supprimer patient

Fonctionnement et utilité :

Cette interface est conçue pour permettre à la secrétaire de supprimer rapidement et efficacement des patients de la base de données. Elle offre un champ de texte pour saisir le matricule du patient à supprimer et un bouton "Confirmer" pour soumettre la demande de suppression. Une fois le matricule saisi et confirmé, l'interface communique avec la base de données Oracle via JDBC pour exécuter une requête de suppression. En cas de succès, un message de confirmation est affiché, sinon un message d'erreur est présenté si le matricule est introuvable ou s'il y a eu une erreur lors de la suppression

Explication des parties importantes :

Connexion à la base de données:

Utilise JDBC pour se connecter à une base de données Oracle, charge le pilote Oracle JDBC, et établit la connexion avec les informations d'identification appropriées

Interface graphique:

Permet à la secrétaire de saisir le matricule du patient à supprimer

Fournit un champ de texte pour la saisie du matricule et un bouton "Confirmer" pour soumettre la suppression

Suppression du patient:

Lors du clic sur "Confirmer", le matricule saisi est récupéré et une requête SQL **DELETE** est construite pour supprimer le patient de la base de données.

La requête est exécutée, et un message de confirmation est affiché si la suppression est réussie, ou un message d'erreur est affiché en cas de problème ou si le matricule est introuvable

Fichier AjoutRen :

Généralités :

Le fichier AjoutRen.java fait partie de l'interface secrétaire dans le projet de gestion de cabinet médical développé en Java avec Window Builder. Son rôle est de permettre aux secrétaire d'ajouter un rendez-vous pour un patient avec les médecins du cabinet

Fonctionnement et utilité :

Cette interface permet de planifier efficacement des rendez-vous médicaux avec médecin principal ou médecin remplaçant. Elle offre à la secrétaire une manière conviviale de saisir la date, l'heure et le matricule du patient pour chaque rendez-vous. Une fois les informations fournies, la secrétaire peuvent choisir entre le médecin principal ou remplaçant en cliquant sur le bouton correspondant. Cette interface facilite la gestion des rendez-vous dans un cabinet médical en automatisant le processus de planification et en assurant une communication transparente entre le personnel médical et les patients

Explication des parties importantes :

Connexion à la base de données:

Cette partie du code utilise JDBC pour établir une connexion à une base de données Oracle. Elle charge le pilote Oracle JDBC et crée une connexion en utilisant les informations d'identification appropriées.

Interface graphique:

L'interface graphique est créée à l'aide de composants Swing tels que JFrame, JLabel, JTextField et JButton. Elle permet à la secrétaire de saisir la date, l'heure et le matricule du patient pour planifier un rendez-vous avec le médecin principal ou remplaçant.

Ajout de rendez-vous avec le médecin principal:

Lorsque la secrétaire clique sur le bouton "Dr. Principale", les informations saisies sont récupérées.

Une requête SQL INSERT est alors construite pour ajouter le rendez-vous dans la base de données avec le statut médical "P" (pour le médecin principal). En cas de succès, un message de confirmation est affiché.

Ajout de rendez-vous avec le médecin remplaçant:

Lorsque la secrétaire clique sur le bouton "Dr. Remplaçant", les mêmes étapes sont suivies que pour le médecin principal, mais avec un statut médical différent ("R" pour le médecin remplaçant)

Fichier SupprimRen :

Généralités :

Le fichier SupprimRen.java fait partie du module de l'interface patient dans le projet de gestion de cabinet médical développé en Java avec Window Builder. Ce fichier est responsable de l'interface permettant de supprimer un rendez-vous pour un patient spécifique

Fonctionnement et utilité :

Cette application permet de supprimer un rendez-vous enregistré dans la base de données. La secrétaire saisit la date et l'heure du rendez-vous à supprimer. En cliquant sur le bouton "Supprimer", le rendez-vous correspondant est effacé de la base de données. Si la suppression est réussie, un message de confirmation est affiché ; sinon, un message d'erreur est généré.

Explication des parties importantes :

Connexion à la base de données:

Cette partie du code utilise JDBC pour établir une connexion avec une base de données Oracle. Elle charge le pilote Oracle JDBC et crée une connexion en utilisant les informations d'identification appropriées.

Interface graphique:

L'interface graphique est construite à l'aide de composants Swing tels que JFrame, JLabel, JTextField et JButton.

Elle permet à la secrétaire de saisir la date et l'heure du rendez-vous à supprimer.

Suppression du rendez-vous:

Lorsque l'utilisateur clique sur le bouton "Supprimer", les informations saisies sont récupérées. Ensuite, une requête SQL **DELETE** est construite pour supprimer le rendez-vous correspondant de la base de données.

Si la suppression est réussie, un message de confirmation est affiché. Sinon, un message d'erreur est affiché pour informer la secrétaire que le rendez-vous spécifié n'a pas été trouvé dans la base de données.

Fichier AfficherRen :

Généralités :

Le fichier AfficherRen.java fait partie du module de l'interface secrétaire dans le projet de gestion de cabinet médical développé en Java avec Window Builder. Ce fichier est responsable de l'affichage de tous les rendez-vous.

Fonctionnement et utilité :

Ce fichier est essentiel pour permettre à secrétaire les rendez-vous prévus pour un patient donné. Lorsque l'application est lancée, cette interface est utilisée pour afficher tous les rendez-vous spécifique.

Explication des parties importantes :

Connexion à la base de données:

Cette partie du code utilise JDBC pour établir une connexion avec une base de données Oracle. Il charge le pilote Oracle JDBC et crée une connexion en utilisant les informations d'identification appropriées.

Initialisation de l'interface graphique:

L'interface graphique est configurée à l'aide de composants Swing tels que JFrame, JLabel, JTable et JScrollPane. Elle affiche une liste des rendez-vous dans une table avec des colonnes pour le numéro de rendez-vous, la date, l'heure, le nom du patient et le médecin. Chargement des données depuis la base de données:

La méthode loadData() est appelée pour exécuter une requête SQL qui récupère les informations sur les rendez-vous depuis la base de données. Les résultats sont ensuite traités et ajoutés au modèle de tableau (DefaultTableModel) pour affichage dans la table.

Gestion des erreurs:

En cas d'erreur lors du chargement des rendez-vous depuis la base de données, une fenêtre de dialogue est affichée pour informer l'utilisateur de l'erreur. Cela garantit une meilleure expérience utilisateur en cas de problème de connexion ou de requête.

Fichier AffichPatient :

Généralités :

Le fichier AffichePatient.java fait partie du module de l'interface secrétaire dans le projet de gestion de cabinet médical développé en Java avec Window Builder. Ce fichier est responsable de l'affichage de tous les Patients

Fonctionnement et utilité :

Ce fichier permet au secrétaire de voir tous les patients enregistrés. Quand l'application est lancée, elle affiche une table avec les informations de chaque patient, comme le matricule, le nom, l'adresse, et d'autres détails. Cela facilite la gestion des informations des patients.

Explication des parties importantes :

Connexion à la base de données:

Cette partie du code utilise JDBC pour établir une connexion avec une base de données Oracle. Il charge le pilote Oracle JDBC et crée une connexion en utilisant les informations d'identification appropriées.

Initialisation de l'interface graphique:

L'interface graphique est configurée à l'aide de composants Swing tels que JFrame, JTable et JScrollPane. Elle affiche une liste des patients dans une table avec des colonnes pour le matricule, le nom, le prénom, l'email, le genre, les maladies, l'adresse, le numéro de téléphone et la date de naissance

Chargement des données depuis la base de données:

La méthode loadPatientData() est appelée pour exécuter une requête SQL qui récupère les informations sur les patients depuis la base de données. Les résultats sont ensuite traités et ajoutés au modèle de tableau (DefaultTableModel) pour affichage dans la table

Gestion des erreurs:

En cas d'erreur lors du chargement des données depuis la base de données, une exception est capturée et les détails de l'erreur sont affichés. Cela permet de diagnostiquer les problèmes de connexion ou de requête, améliorant ainsi la robustesse de l'application.