

COMPTE RENDU PROJET : GESTION HOPITAL

Description de contexte

Le sujet :

Notre application a pour but de pouvoir gérer le fonctionnement d'un hôpital de la prise d'un rendez-vous, au traitement, en passant par la création de fiche médical et d'ordonnance. Elle a pour but d'être simple, rapide et d'être la plus réaliste possible.

Pourquoi ce projet :

Nous avons choisi de développer cette application de gestion car il s'agissait d'un sujet concret et dont nous pouvions nous inspirer de la vie réel. De plus le fait de pouvoir développer un logiciel exploitable et disposant de multiple possibilité de fonctionnalités a été le point nous déterminant à faire ce projet.

Les fonctionnalités que nous avons prévues :

Un système de connexion permettra d'identifier les médecins, le personnel ainsi que l'administrateur. Le programme est initialement prévu pour être utilisé par les personnes qui vont réceptionner les patients, ou les futurs patients. Il doit donc y avoir une interface pour ajouter des patients mais aussi pour leur créer une fiche médicale (la fiche sur laquelle sont inscrites les informations concernant le patient qui seront utile au médecin qui le prendra en charge tel que les symptômes donc souffre le patient). Un patient peut avoir plusieurs fiches médicales : il peut souffrir d'une grippe en 2017 et revenir en 2020 pour une bronchite. En revanche une seule fiche médicale est active à la fois, les fiches devenues inactive doivent être archivées. Une fois la fiche médicale créer, un rendez-vous peut être créé avec un médecin. A l'issue de ce rendez-vous, une ordonnance doit pouvoir être générée en fonction des symptômes du patient. Evidemment, l'avis du médecin est le plus important, il doit donc vérifier cette ordonnance qu'il pourra modifier à sa guise, et rajouter ses instructions avant de la valider. Une fois l'ordonnance validée, le patient pourra la consulter à sa guise à l'aide de son numéro de fiche médicale. L'administrateur pourrait quant à lui ajouter des médecins, des personnels hospitaliers (des infirmiers aidant les médecins par exemple), s'occuper des médicaments utilisés au sein de l'hôpital et pourrait gérer toute les bases de données de l'hôpital. Nous imaginons également un système de répartition par service des patients et des médecins ainsi que de possibles machines associés à chacun des services.

Description des vues



The login screen for CHU Auvergne features a blue background. At the top center is the text "CHU Auvergne". On the left, there is a text prompt "Patient dans notre CHU ? entrez votre n° Dossier ici" above a white input field. To the right of this, there are two more input fields: "Nom d'utilisateur :" and "Mot de passe :". Below the first input field is a small "OK" button. At the bottom center is a large "Connexion" button.

Vue permettant de se connecter en tant qu'administrateur, médecin et agent d'accueil via un nom d'utilisateur et un mot de passe. Un patient du CHU peut aussi consulter son dossier le numéro de ce dernier.



The administrator dashboard for CHU Auvergne has a blue background. At the top center is the text "CHU Auvergne". In the top right corner is a "Deconnection" button. The main area is divided into two columns. The left column contains two large rectangular buttons: "RENDEZ-VOUS" (top) and "FICHES MEDICALES" (bottom). The right column contains three smaller rectangular buttons stacked vertically: "GESTION PERSONNEL", "GESTION MEDICAMENT", and "GESTION SYMPTOME".

Vue administrateur, un administrateur peut gérer les rendez-vous, regarder les fiches médicales, gérer le personnel, gérer les symptômes, gérer les médicaments et voir l'ordonnance d'un patient.



Vu Medecin, il dispose de la liste de ses rendez-vous et peut « créer l'ordonnance » correspondante au rendez-vous selectionné.



Vu d'un Personnel simple.

[illegible]

Tous les acteurs (sauf le patient) peuvent ajouter, modifier ou supprimer un rendez-vous via cette vue.

Retour

LISTE FICHES MEDICALES

Deconnection

Celine Merle - 013012017CM

jhg jknk - 013012017jj

Ajouter

Supprimer

Vue permettant de créer une nouvelle fiche médicale ou d'en supprimer une déjà existante.

Retour **GESTION PERSONNEL** **Deconnection**

NOM :

PRENOM :

MOT DE PASSE :

MEDECIN : ☐

Supprimer **Ajouter**

Vue permettant d'ajouter ou de supprimer un membre du personnel du CHU, si la case médecin est cochée, le nouveau membre sera considéré comme médecin sinon, il sera considéré comme agent d'accueil.

Retour **AJOUT SYMPTOME** **Deconnection**

migraine
fracture
paralysie
nausée
hematome
dolipran

NOM :

DESCRIPTION :

Ajouter

Vue permettant d'ajouter un nouveau symptôme mais pas d'un supprimer un, on considère qu'un symptôme étant apparu une fois doit conserver dans la base de donnée.

sd
sanfond
eau benite

Mr Fauret

Indications pour : sd
Indications pour : sanfond
Indications pour : eau benite

<-- MEDICAMENTS

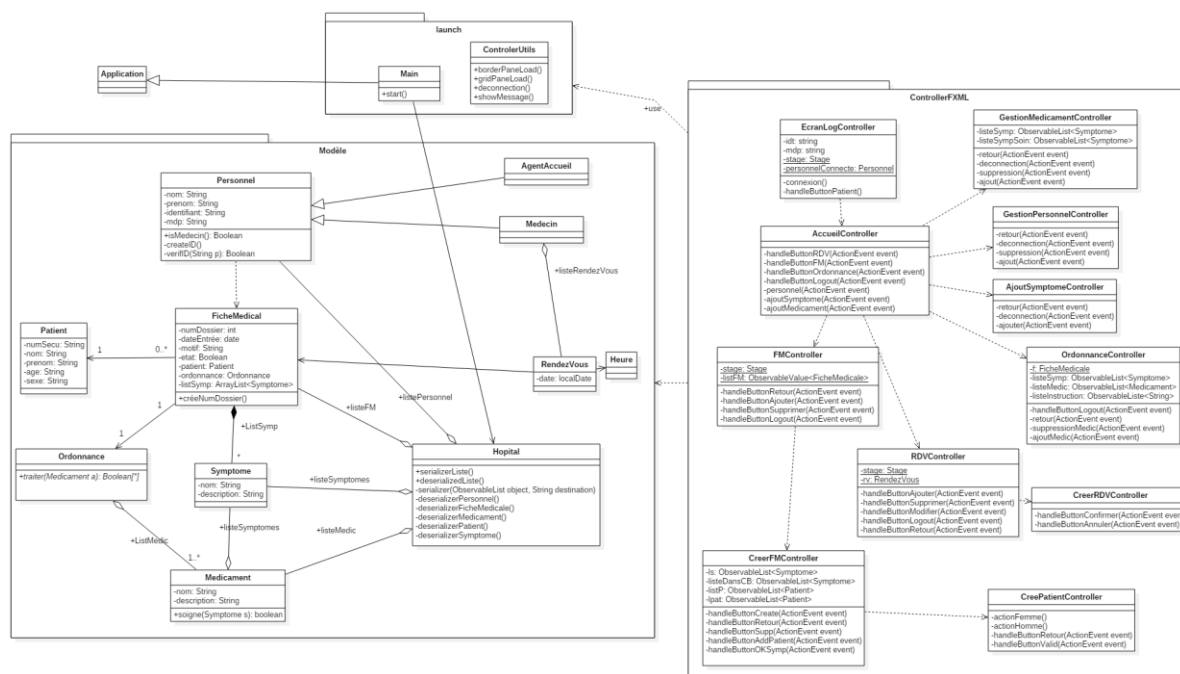
Si vous etes guérris, n'hesitez pas à en informer otre Medecin traitant. Si votre etat empire ou reste inchangé pendant les 15jours suivant le rendez-vous, contacter votre Medecin

INSTRUCTIONS -->

QUITTER

Vu de l'ordonnance que le patient peut consulter a tout instant

DIAGRAMME DE CLASSE



Description de l'architecture et du diagramme de classe :

Dans cette application nous suivons une architecture MVC (Modèle-Vue-Contrôleur) et disposons d'une couche persistance appelé data permettant l'enregistrement de l'ensemble des listes du programmes de gestion ainsi que leur chargement lors du lancement de l'application. Un package ressources est également visible et nous permet de stocker la partie non essentiel du programme c'est-à-dire les ressources CSS que nous utilisons (background etc...).

Le Modèle :

Classe Patient :

La classe Patient contient les attributs permettant le remplissage de la fiche médicale associé à celui-ci. Pour chaque patient on dispose d'une et unique fiche médicale qui est mise à jour à chaque nouveau rendez-vous. L'ensemble de ces attributs sont récupéré dans la classe fiche Médical notamment pour pouvoir associé nom et prénom d'un patient à une fiche médical.

Classe Fiche Médicale :

Cette classe regroupe l'ensemble des attributs nécessaires dans le cadre de l'hôpital ainsi qu'un numéro de dossier unique à chaque fiche médical qui est créé directement dans la classe lors de la création de la fiche médical. Cette méthode de création prend en compte

l'âge du patient. Une liste de fiche médicale est notamment stocker au sein de la classe hôpital permettant à partir du numéro de dossier une recherche de celle-ci. Le numéro de dossier permet également au patient de se connecter via son numéro de dossier et d'avoir ainsi accès à sa dernière ordonnance d'où la liaison directionnel vers la classe ordonnance permettant de récupérer l'ordonnance. Une fiche médicale dispose également d'une liste de symptôme associé, cela permet notamment de trouver les médicaments associés pour pouvoir par la suite créer une ordonnance et soigner le patient.

Classe Rendez-Vous :

Cette classe utilise la fiche médical récupérant ainsi les informations (nom, prénom du patient) nécessaires. Elle est également lié à personnel pour récupérer le nom et le service du médecin avec qui le patient à rendez-vous. Elle dispose d'un attribut Heure qui lui est passé par la classe du même nom qui lui donne un ensemble d'heure ouvrable. Elle dispose également d'un attribut date pour fixer la date voulue pour le rendez-vous

Classe Personnel :

Un personnel hospitalier dispose des attributs nécessaires à son identification (nom et prénom permettant de former l'identifiant), les personnels ont la possibilité de se loguer pour avoir accès aux fonctionnalités de l'hôpital. Un personnel peu créer une fiche médicale ou en rechercher une existante elle utilise pour cela la classe du même nom. Ces deux méthodes permettent à la création d'un nouveau personnel par l'administrateur de créer un identifiant et de le vérifier lorsqu'un personnel hospitalier se logue.

Classe Médecin :

La classe médecin hérite de l'ensemble des attributs de la classe personnel ainsi que de l'ensemble de ces méthodes dont la méthode isMedecin qu'on fixe à true pour identifier le statut. A cette classe on passe une liste de rendez-vous permettant au médecin de voir tous les rendez-vous qui lui son affilié.

Classe Accueil :

Cette classe hérite simplement de personnel et de ces méthodes.

Classe Symptômes :

Composé simplement d'attribut elle définit ce qu'est un symptôme.

Classe Médicament :

La classe médicament prend une liste de symptôme en attribut car chaque médicament a pour but de soigner cette liste. La méthode soigne permet de passé l'état d'une fiche médicale à true pour indiquer que le patient est traité et que la fiche médicale peut être archivée. Une liste de médicament compose l'ordonnance d'où la liaison entre les deux.

Classe Ordonnance :

La classe ordonnance dispose d'une liste de médicament qu'elle récupère auprès de la classe du même nom. Elle dispose également d'une méthode qui permet de passer l'état d'une fiche médicale à soigner d'où l'utilisation de l'attribut de la classe.

Classe Hôpital :

Ensemble des listes de personnel, médicaments, symptômes et fiche médicale formant une base de données. Le Main récupère ensuite toutes ces listes et les donnent au contrôleur ce qui leur permet de les manipuler et de les afficher.

Le ControllerFXML :

Classe EcranLogController :

Elle dispose de la méthode connexion qui va permettre d'identifier un personnel ou un numéro de dossier en récupérant la liste de personnel dans le Main et vérifié ainsi que le personnel existe bien. Le handleButton permet simplement de récupérer une ordonnance associé au numéro de dossier passé dans le champ de connexion du patient.

Classe OrdonnanceController :

La classe récupère simplement la liste de médicament dans le Main pour pouvoir l'afficher. Le handleButtonLogout sert simplement à activer la déconnexion et à revenir a la page de connexion.

Classe RDVController :

L'ensemble des handle ont pour but d'ajouter, modifier ou supprimer un rendez-vous lors l'action bouton. Ajouter va ainsi ouvrir une nouvelle fenêtre pour la création du rdv tout comme modifier. Le handle retour permet simplement d'afficher la vue précédente.

Classe AjoutSymptomeController :

Cette classe disposant d'une méthode handle ajouté qui permet d'ajouter un symptôme à la liste de ceux-ci. On ne peut pas supprimer de symptôme puisque dans la logique de l'application lorsqu'un symptôme à été relevé un moment t il doit être conservé dans le cas où il réapparaîtrait un jour.

Classe AddFMController :

Permet simplement la création d'une nouvelle fiche médicale en faisant apparaitre la fenêtre de remplissage de la fiche. Les handle Patient et OKsymp permettent respectivement de sélectionner un patient et de confirmer les symptômes de celui-ci finalisant ainsi la création de la fiche.

Classe FMController :

Gère l'interface qui supprime/modifie ou ajoute dans la liste de fiche médical d'où une nouvelle fois la liaison avec le Main qui permet de récupérer cette liste.

Classe GestionPersonnelController :

Permet l'ajout ou la suppression au moment du clic bouton dans la liste des personnels (récupéré en attributs depuis le Main).

Classe GestionMedicamentController :

Même action que le gestionPersonnelController mais sur la liste de médicaments.

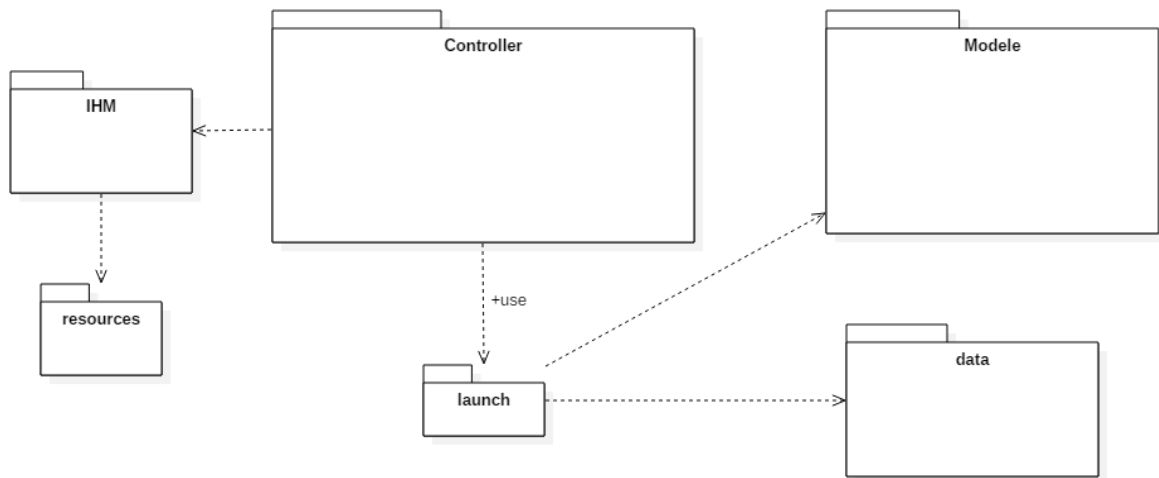
Classe CréePatientController :

Cette classe est associée à la gestion des champs du formulaire de création d'un patient permettant de spécifier son sexe (actionHomme, actionFemme) et remplir l'ensemble des champs et d'en vérifier le contenu (remplie ou non) puis permet l'ajout d'un nouveau patient à la liste.

Classe SelectFMController :

Gère la création d'un rendez-vous avec sélection d'une fiche médicale associé.

Diagramme de paquetage Gestion d'hôpital :



Documentation diagramme de package :

Le package controller :

Il regroupe l'ensemble des contrôleurs FXML utiles à la manipulation des vues de l'application. Il récupère les données présentes dans « data » via le Main et utilise des fonctions du package launch.

Le package modèle :

Il s'agit de l'ensemble de la couche métier de l'application ou sont regroupées toutes les méthodes permettant la fonctionnalité du programme. Dans le modèle, les listes composant l'application sont initialisées et mises à jour, puis récupérées par le Launch.

Le package Launch :

Le launch sert de package d'enregistrement et de chargement des données de l'application d'où sa communication avec le package data qui regroupe les données enregistrées de l'application. En les récupérant, il peut par la suite passer celle-ci au contrôleur qui pourra les manipuler.

Le package resources :

Il est simplement utilisé par le package IHM (les vues) pour pouvoir récupérer les éléments de style qui le composent (background, image etc...).

Note : Les détails des paquets sont visibles au sein du diagramme de classe.