

RAPPORT DE PROJET



CONCEPTION D'UNE APPLICATION POUR LA GESTION D'UN « HOPITAL »

Elaboré par :

* Zied Khayechi

2017-2018

Description de contexte

Notre application a pour but de pouvoir gérer le fonctionnement d'un hôpital (ou cabinet médical) de la prise d'un rendez-vous, au traitement, en passant par la création de fiche médical et d'ordonnance. Outre la gestion d'un hôpital, elle permettra aux patients de consulter leur ordonnance sous forme numérique.

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 utiles au médecin qui le prendra en charge tel que les symptômes dont 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. Une fois la fiche médicale créée, un rendez-vous peut être prévu avec le 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. La fiche correspondante a alors fait l'objet d'un traitement par le médecin et devient inactive, mais les fiches médicales doivent être archivées afin de garder un historique des problèmes de santé du patient. L'administrateur pourra 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.

Les fonctionnalités mise en œuvre

Il s'agit là d'un compte rendu du travail fait sur l'application :

Le système de connexion / déconnexion est fonctionnel.

Actuellement l'administrateur peut ajouter des symptômes : de nouvelles maladies peuvent apparaître, il est donc essentiels de pouvoir ajouter les symptômes qui ne sont pas encore existant.

Il peut également ajouter ou supprimer du Personnel, tel que des médecins ou de simples employés.

Il peut ajouter des Médicaments, comme pour les symptômes, on ne connaît pas les médicaments du futur, il faut donc pouvoir en rajouter. Il peut aussi supprimer des médicaments : on s'aperçoit que certains médicament son nocif et sont retirés du commerce.

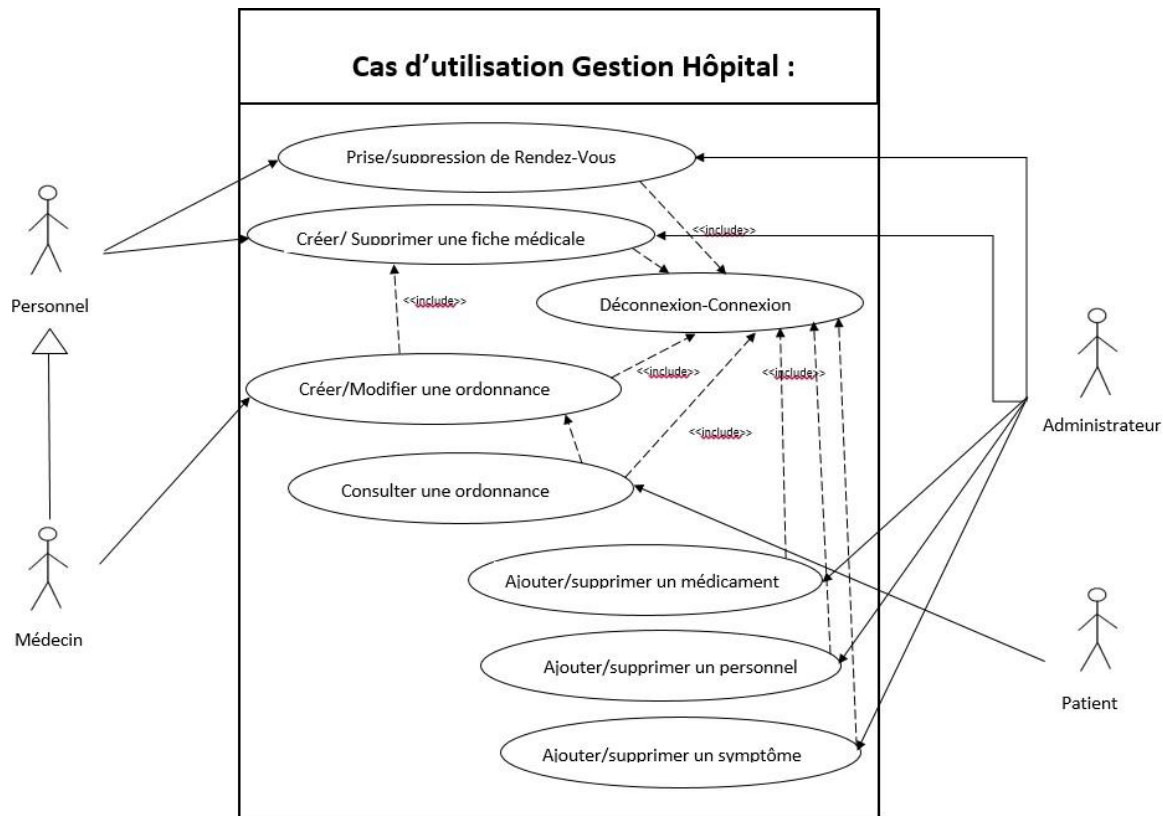
Il y a la possibilité pour tout le personnel hospitalier d'ajouter des patients, de leurs créer des fiches médicales et prévoir des rendez-vous avec les médecins.

Sur l'écran d'accueil, le médecin connecté peut voir tout ses rendez-vous avec la date. Il peut, suite à ces rendez-vous, générer une ordonnance : elle propose tous les médicaments adaptés afin de soigner les symptômes inscrits sur la fiche médicale. Le médecin pourra ajouter des instructions, et ajuster les médicaments s'il juge que l'ordonnance proposé par l'application n'est pas optimale.

Une fois l'ordonnance créer, le patient peut a tout moment la consulté a l'aide de son numéro de fiche médicale.

Toute les Fiches médicales créées sont archivées, mais il n'y a pas d'interface pouvant faire l'historique d'un patient.

Certaines données telles que la description d'un médicament, celle d'un symptôme, ne sont pas utilisées : elles sont là pour ne pas avoir à être implantée si le logiciel venait a être amélioré dans le futur.



Documentation :

Personnel : L'acteur d'accueil a la possibilité si il est connecté de prendre un rendez-vous pour un patient en cliquant sur le bouton « Rendez-vous » puis « ajouter ». L'agent crée alors une nouvelle fiche médicale. Dans cette fiche médicale l'agent d'accueil remplit le nom, prénom, l'âge ainsi que l'ensemble des autres champs (il peut également modifier ultérieurement celles-ci voir la supprimer si besoin). Une fois la feuille remplie, un rendez-vous est noté dans la liste, l'acteur saisie alors la date, l'horaire et le médecin concerné. Un agent d'accueil peu modifier un rendez-vous en sélectionnant celui-ci puis en cliquant sur « modifier » qui lui ouvre une page ou il peut changer la date, l'horaire et le médecin. Il peut enfin supprimer un rendez-vous de la liste en cliquant sur « supprimer » (une vue de confirmation et demandé lors de cette action).

Patient : Un patient en saisissant son numéro de dossier en tant qu'identifiant sans aucun mot de passe peu accéder à son ordonnance la plus récente.

Médecin : Un dispose des actions d'un Personnel en ce qui concerne la prise de rendez-vous et la création de fiche médical. Mais il a également la possibilité de modifier une fiche médicale en y ajoutant des symptômes. Ces symptômes s'ajoute à la fiche suite au clic du

médecin sur « ordonnance » qui ensuite saisie les observations effectués sur le patient. Le clic par la suite sur le bouton « confirmer » permet la création de l'ordonnance préfète (liste de médicaments associés à chaque symptôme saisi) qui va s'afficher dans une nouvel vue. Dans cette vue le médecin peu décocher certain médicament mais également en rajouter en sélectionnant ceux-ci dans une liste modifiant ainsi l'ordonnance puis clic sur « confirmer » ou « supprimer ». S'il fait l'action de suppression il retourne alors au menu de base. Il peut également consulter les ordonnances de ses patients en fonction de leur nom et obtient ainsi la liste de toutes les ordonnances pour ce patient. Il peut ainsi la re-sélectionner pour directement la réimprimer par exemple dans le cadre d'un traitement sur la durée.

Administrateur : L'administrateur lorsqu'il se connecte est un personnel a part disposant des mêmes actions que ceux-ci mais qui a la possibilité de gérer les personnels de l'hôpital, ainsi que les symptômes et médicaments connu et présent dans celui-ci

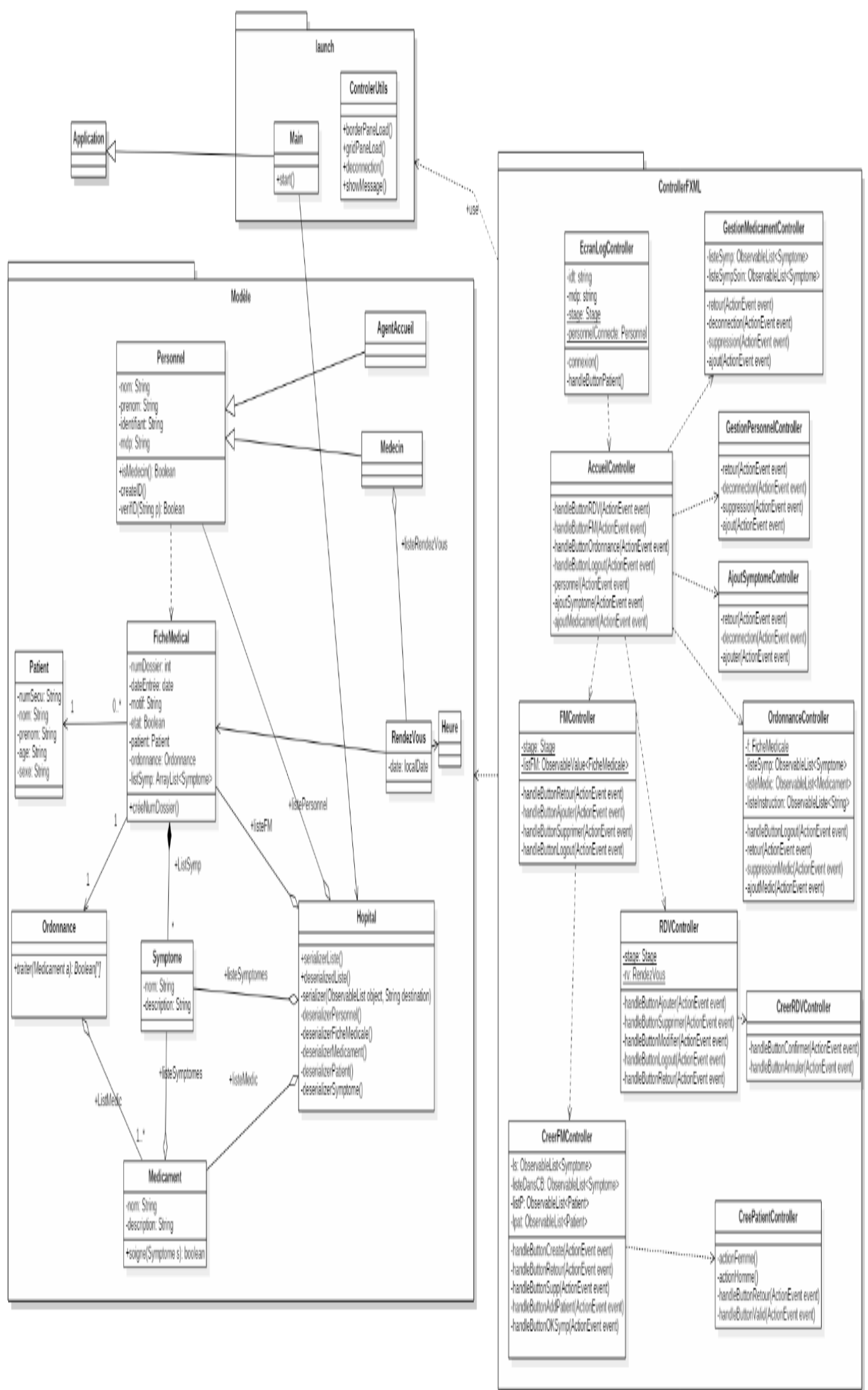


DIAGRAMME DE CLASSE

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).

Le Modèle :

Classe Patient :

La classe Patient définit les informations du patient de l'hôpital en tant qu'humain : c'est à dire son nom, son prénom, son âge et son n° de sécurité social.

Classe Fiche Médicale :

La fiche Médicale d'un patient contient les informations qui seront nécessaire au médecin.

Un numéro unique à la fiche est créé lors de la création de cette dernière.

L'état de la fiche passe de 0 à 1 lorsque le patient est considéré comme guérit (dans l'algorithme, c'est le moment où il reçoit son ordonnance).

Elle possède aussi l'ordonnance, qui sera ajoutée à l'issue du rendez-vous avec le médecin.

Classe Rendez-Vous :

Cette classe contient la fiche médicale concernée, ainsi que le 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 basiques de l'hôpital.

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` servant à indiquer si le Personnel est un médecin ou non. Il possède une liste de rendez-vous contenant tous les rendez-vous qui lui son affilié.

Classe Symptômes :

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

Classe Médicament :

La classe médicament prend une liste de symptôme qu'il soigne en attribut. La méthode `soigne` permet de savoir si le symptôme passé en paramètre est soigné par ce médicament. Une liste de médicament compose l'ordonnance d'où la liaison entre les deux.

Classe Ordonnance :

Cette classe possède un dictionnaire : à chaque Médicament inscrit sur l'ordonnance correspond une instruction donnée par le médecin.

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.

Elle permet aussi a un patient d'entrer son numéro de Fiche médicale pour avoir accès a son ordonnance.

Classe OrdonnanceController :

Génère l'ordonnance.

Classe RDVController :

Affiche la liste des Rendez-vous afin d'en ajouter, ou en supprimer.

Classe AjoutSymptomeController :

Permet l'ajout d'un Symptome à la base de données.

Classe CreerFicheMedicaleController :

Permet de créer une nouvelle fiche médicale.

Classe FMController :

Affiche la liste des Fiches Médicales en cours de traitement.

Classe GestionPersonnelController :

Permet l'ajout ou la suppression de Personnel dans la liste des personnels.

Classe GestionMedicamentController :

Même Permet l'ajout ou la suppression de Médicament dans la liste des médicaments.

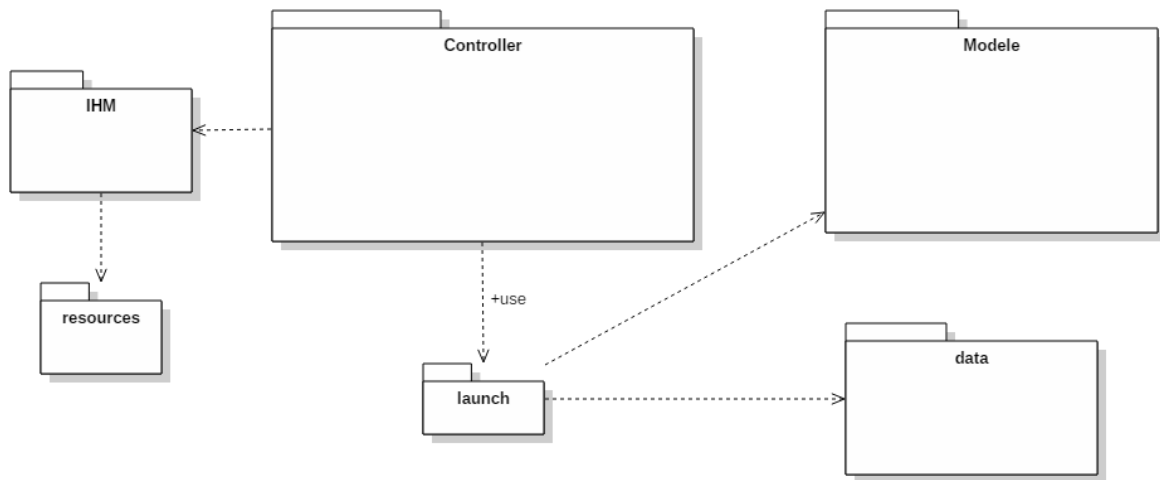
Classe CrééPatientController :

Permet la création d'un patient

Classe CreerRDVController :

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 utile 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 son regroupé toutes les méthodes permettant la fonctionnalité du programme. Dans le modèle les listes composant l'application sont initialisé et mise à jour puis récupérées par le Launch.

Le package Launch :

C'est ici que se lance l'application.

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 controller 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 compose (background, image etc...).

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

connexion en admin: login/Mot de passe :admin

