**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 était relevé un moment t il doit être conservé dans le cas où il réapparaitrait 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.