



## Projet : HOPITAL : gestion d'un hôpital et de ses patients v1

(option :Réaliser vos diagrammes de classes avant de coder si vous le souhaitez !!)

### Partie 1 (métier):

Un hôpital est constitué de sa secrétaire et de la file d'attente de ses patients

(hopital unique-Singleton)

les patients arrivent au fil de l'eau afin de passer en consultation chez un des médecins d'une des 2 salles

Les patients sont censés remplir un formulaire de données personnelles

Les patients sont censés être connus dans les systèmes de l'hôpital pour ensuite prétendre être rajoutés à la file d'attente

Il existe des médecins, chacun dans une salle (la salle 1 ou la salle 2)

Une interface utilisateur (en mode console) permet de se positionner en tant que médecin de la salle 1 ou salle 2 ou en tant que secrétaire, en passant par le module d'authentification

### module authentification

il permet de s'authentifier en tant que secrétaire ou médecin

grâce au login et password depuis la table\_authentification, le système permet de proposer un menu cohérent, ou de refuser l'accès

si la personne qui tente de se connecter au système est bien identifiée, on récupère alors son rôle dans l'hôpital (0 secrétaire / 1 ou 2 médecin) et puis son nom

### Les opérations de la secrétaire :

1. Rajouter à la file d'attente un patient :
  - a. si existant, dans ce cas il faut le récupérer de la base
  - b. s'il est nouveau il faudra l'inscrire en base
2. Afficher la file d'attente
3. Afficher le prochain patient de la file (sans le retirer)
4. Sortir de ce menu et revenir au menu principal

### Les opérations du médecin:

Le médecin occupe une salle, la 1 ou la 2 en fonction de son affectation

1. le médecin décide de rendre la salle dispo

(il fait quitter le patient actuel et le prochain patient arrive **automatiquement OBSERVER**)

On affichera alors les infos du nouveau patient arrivé

2. Afficher la file d'attente

Chaque patient dans une salle constitue une visite (voir détail plus bas)

au fur et à mesure des visites, les visites sont stockées dans une liste de visite, afin d'être inscrit a terme en base dans la table des visites

(à l'initiative du médecin, ou lorsqu' on a atteint la taille max d'un lot à savoir 10 visites par exemple)

3. le médecin peut décider de sauvegarder en base les visites (exemple fin de journée)

4. il peut quitter le menu

## **Partie 2 (technique):**

Le module lanceur (main), doit proposer un menu global et sous menu, intelligent, permettant de proposer toutes les possibilités et de revenir au menu précédent

Un patient : id,nom ,prenom,age mais aussi téléphone et adresse

Une Visite : numéro de visite (auto),Id du patient (numéro de sécu-unique), nom du médecin ,le numero de salle, le cout de la visite (tarif fixe=23 e ) ainsi que la date et heure de la visite (une chaine ou une date)

Une salle possède un numéro de salle (1 ou 2), médecin associé

Un DAOPatient ,DAOVisite et , DAOAuthentification sont nécessaires pour faire fonctionner la connexion a la base hospital

## **Design Pattern**

L hospital est unique (singleton), Observer

## **Partie 3 : modules complémentaires**

Le médecin doit pouvoir afficher la liste de ses visites en base :

La secrétaire permet d'afficher, pour un patient la liste de ses visites en base, selon son identifiant

On écrira dans un fichier texte la date et heure d'arrivée pour chaque patient qui entre dans la file d'attente

(Il faudra voir comment fait-on pour récupérer la date et heure et comment enregistrer ces infos dans un fichier .txt)

Ex : hospital.txt :

1 toto titi 17/01/2023 10h35

3 x y 17/01/2023 10h45

5 aa bb 17/01/2023 12h00

## **Partie 4 partage des taches :**

Le partage des taches peut se faire en deux ou trois modules qui sont :

- 1 .Module principal : le main avec la logique de menu et sous menu, sans oublier que ce module est l'intégrateur des autres
2. module d'accès a la base HOPITAL, les DAO et les java bean (pojo)
3. classes de traitements, gestion de la file d'attente....

### Partie 5 Les bases de données :

Base hopital-obs

Les tables :

patients---id, nom, prenom, age,telephone , adresse [alimenté par l'application java]

visites- id(AI), idpatient, date, médecin, num-salle, tarif(23) [alimenté par l'application java]

authentification :login , password, nom, métier(0-secretaire, 1 médecin salle1, 2 médecin salle2)

[Alimenté depuis le workbench...saisir qq exemples comme vu ensemble lors de la démo]

Nb :

La démo est a titre indicatif

Vous pouvez proposer votre propre logique, mais les grandes lignes sont les mêmes

Quelques options supplémentaires pourront être rajoutées plus tard...

Le projet est a faire en équipe(3 personnes), vous devez fixer le partage avant de démarrer,

Il est souhaitable de valider de bout en bout les modules au fur et a mesure

Une démo fonctionnelle est souhaitable par équipe

BON COURAGE ☺ (pour la v1)