Dossier technique du projet de C++

Ce dossier technique vise à présenter notre projet de système de gestion des patients dans un centre de santé. Il spécifie les différentes classes appliquées dans le projet, les méthodes qui les composent ainsi que les attributs de chacune d’entre elles. Il présente également les relations qu’il y a entre chacune de ces classes.

*    **Feuille médicale** (élément de la classe Patient)

Elément de type struct :

·       dateHeure : DateTime ;

·       nom Médecin de la consultation : string ;

·       Diagnostique : string ;(peut servir à stoquer l’état de santé initiale)

·       factures : double ;

*    **Consultation**

. recupére un patient

. le place en tete de la liste de consutation pour une prise en charge rapide

*    **Facture**

·       montant : double ;

·       dateEmission : DateTime ;

·       etatPaiement : bool (payé ou non)

*    **Rappel**

Elément de type pile(stack) :

·       dateHeure du rappel : DateTime ;

·       message du rappel : string ;

***Presentation des classes du projet :***

1. **classe Patient**

Attributs privés :

* dossierMedical : élément de type vector de variant pouvant être un string , un double, un int, un char  et une feuille médicale
* nom : string (élément du cahier médical)
* prenom : string (élément du cahier médical)
* age : int (élément du cahier médical)
* sexe : char (élément du cahier médical)
* matricule : string (identifiant unique servent à l’authentification d’un patient)
* dateAdmission : Date (automatiquement récupéré sur le pc)
* factures : double(represente la somme des factures de chaque consultations)

**Constructeur** :

Initialise un nouveau dossier médical avec les informations personnelles et l’état de santé initial.

**Méthodes** :

Definirmatricule() :pour se voir assigner un numero d’identifiant unique de façon automatique

Consulterdossiermedical() :lire son dossier medical pour suivre les diagnostiques qui lui ont été recommendés

consulterRappels() : renvoir les rappels qui ont été envoyés au client

supprimerRappels() : supprime les rappels que le patient trouve encombrant

consulterFacture () : renvoie la facture que dois le client en tout au centre de santé

1. **classe Medecin**

**Relation** : Ami(friend) de la classe Patient (accès à tous les membres privés et protected de la classe patient)

Constructeur : marque les nom du médecin

**Méthodes** :

envoyerRappel(patient : Patient, dateHeure : DateTime, message : string ) :les rappel insère un texte pour rappeler au patient un évènement un message par défaut est envoyé si aucun texte n’est entré

consulterDossier (patient : Patient) :opteneir l’acces total au dossier du patient pour le consulter lire des diagnostiques faits par d’autres medecins

poserDiagnostique: modifie le dossier médical du patient  pour y ajouter des informations du diagnostic(des consultations des médicaments recommandés etc..) et ensuite marque la facture de la consultation

planifierRDV (patient : Patient, dateHeure : DateTime) :utililisable par le personnel administratif

1. **classe PersonnelAdministratif**

**relation :  hérite de médecin application du principe du polymorphisme en redefinisant certaines méthodes pour éviter l’acces de façon claire aux dossiers des patients**

**constructeur : utilise celui de la classe medecin**

**Méthodes spécifiques :**

gererFactures (patient : Patient, paiement : double) : marque une facture d’un patient payée

genererFichierOrdonnance (patient : Patient, ordonnance : string) : sauvegarde les données des patients de la liste des patients dans des fichiers .txt pour des utilisations ultérieures

enregistrerPatient (fichierPatient : File) : marque un patient comme appartenant à la liste de patient de l’hôpital en l’insérant dans la liste\_patient qui est une liste de classe patients

plannifierRDV( patient : Patient, dateHeure : DateTime) :positionne le patient qui veut un rendez vous en tete de la liste de patient il gere les cas d’urgence

**5.fichier des includes**

**Contient les includes qui seront utilisé dans la plupart des fichiers comme iostream**



**Relations clés :**

✅ **Patient** contient une liste chaînée DossierMedical  
✅ **Patient** a plusieurs Consultation (historique de diagnostics avec les infos de la consultation)  
✅ **Patient** contient une **facture à payer** au centre de santé chez le **personnel administratif**  
✅ **Medecin** est ami de **Patient** (friend)  
✅ **PersonnelAdministratif** hérite de **Medecin** (: public Medecin, héritage public)  
✅ **MenuPrincipal** gère la **connexion**