



Projet NF06

Mini base de données de gestion hospitalière

UE - NF06

Rédacteurs : **Maya Mazuet**
Charlotte Noe

Semestre A23

Finalisé le 15/12/2023

Projet réalisé dans le cadre de NF06.

Implémentation d'un mini base de données de gestion hospitalière.



I. Sommaire

I. Sommaire.....	2
II. Manuel d'utilisation.....	2
A. Programme principal en C.....	2
1. Saisie d'un patient.....	2
2. Saisie d'un médecin.....	3
3. Rechercher un dossier patient.....	3
4. Affichage des dossiers t.....	3
5 et 6. Suppression et modification d' un dossier patient.....	3
B. Partie statistique en Python3.....	4
III. Explication du code source.....	4
A. Programme principal en C.....	4
1. Le main.....	4
2. Fonction "compte".....	5
3. Fonction "dernier_num".....	5
4. Fonction "recherche".....	5
5. Fonction "afficher".....	5
B. Partie statistique en Python3.....	5

II. Manuel d'utilisation

A. Programme principal en C

Le programme débute avec un menu qui propose à l'utilisateur de choisir une des fonctions parmi les 6. Si l'utilisateur rentre - 1 le programme s'arrête. Seul un entier doit être rentré par l'utilisateur. En cas d'une valeur non conforme, la machine redemandera une nouvelle valeur à l'utilisateur. L'utilisateur peut immédiatement utiliser toutes les fonctionnalités car des dossiers sont déjà enregistrés.

1. Saisie d'un patient

L'utilisateur choisit entre une consultation ou une urgence puis remplit les informations selon les consignes données par la machine. Les informations demandées sur les patients seront les mêmes dans les 2 cas.

Lors de la saisie des informations, si l'utilisateur ne respecte pas les contraintes, la machine redemandera l'information en précisant les problèmes (exemple : une

date non conforme etc.). Cependant, l'utilisateur doit respecter le type (si un nombre est demandé, il ne doit pas rentrer un caractère).

La description de la maladie et les antécédents du patient peuvent être des phrases. Un retour à la ligne annonce à la machine de passer à l'information suivante.

Si le médecin attribué n'existe pas, il ne peut pas être attribué. La machine affichera les médecins déjà enregistrés de l'hôpital.

Le nombre de chambres est limité, si une chambre est déjà attribuée, le programme redemandera une nouvelle valeur.

Une fois les informations rentrées avec succès, le programme affiche le devis total qui comprend le coût de l'examen et le coût du nombre de nuits dans une chambre individuelle.

2. Saisie d'un médecin

Cela permet à l'utilisateur d'ajouter un nouveau médecin à la base de données de l'hôpital. L'utilisateur suit les informations demandées par la machine.

Pour la recherche, l'affichage, la suppression et la modification, l'utilisateur choisit où faire sa requête : soit dans les urgences soit dans les consultations.

3. Rechercher un dossier patient

Il choisit le mode de recherche : recherche par numéro de dossier ou selon le nom complet du patient.

Si le dossier existe, il est affiché à l'écran.

Le numéro de dossier étant unique, un seul dossier sera affiché. Si deux patients ont le même nom et le même prénom, les deux dossiers seront affichés.

4. Affichage des dossiers

Il choisit le mode d'affichage : tous les dossiers, par ordre alphabétique ou à une date précise.

La date doit être écrite avec des / comme par exemple : "12/12/2020".

5 et 6. Suppression et modification d' un dossier patient

Il choisit ensuite le mode de recherche : soit une recherche par numéro de dossier, soit selon le nom complet du patient.

Si le dossier est trouvé, la machine demande confirmation de la suppression / modification. L'utilisateur peut donc changer d'avis et sera retourné au menu. En cas de recherche par nom, si un nom est attribué plusieurs fois, la machine demandera pour chaque dossier ce que l'utilisateur veut faire.

B. Partie statistique en Python3

Il est possible d'ajouter de nombreuses autres statistiques mais nous avons décidé de nous limiter à 6 possibilités.

III. Explication du code source

A. Programme principal en C

Les structures sont définies dans le fichier header "struct_hopital.h".

Le choix de la taille des chaînes de caractères est un choix arbitraire. Nous avons pris des tailles assez grandes permettant de ne pas limiter l'utilisateur lors de la saisie des informations. Différents éléments ont été ajoutés au projet comme par exemple : l'ajout des médecins dans un fichier "médecins.txt", la demande des antécédents du patient, ou encore la vérification de la disponibilité des chambres.

1. Le main

Il comporte le cœur du programme avec le menu.

Le programme est continu : après chaque requête, l'utilisateur est renvoyé au menu où il peut effectuer une autre requête. Cela permet d'ajouter une certaine fluidité lors de l'utilisation du programme.

Dans la fonction "ajout" et "ajout_medecin", de nombreuses contraintes doivent être respectées. Elles sont vérifiées par la machine grâce à une boucle indéterminée et permettent d'avoir des informations cohérentes. Certains choix ont été pris arbitrairement lors de l'élaboration du programme.

Nous avons décidé de considérer qu'il était envisageable pour l'utilisateur de rentrer une date d'arrivée antérieure à la date actuelle, dans le cas où l'utilisateur voudrait faire une modification sur un ancien dossier, ou dans le cas où un dossier n'aurait pas été rentré à temps (exemple le passage de 23h59 à 00h00). Nous avons choisi 1998 comme date maximale, en supposant qu'il s'agit de la date d'ouverture de l'hôpital.

La saisie des informations des patients est enregistrée dans un fichier texte.

Les données sont séparées par des virgules pour être plus facilement transformées en fichier csv et ainsi utiliser les fichiers pour les statistiques du programme python.

3 fichiers txt distincts permettent de stocker les informations : des urgences, des consultations et des médecins.

2. Fonction “compte”

Elle permet de compter le nombre de clients dans un fichier. Cette fonction est utilisée à plusieurs reprises au sein du programme.

3. Fonction “dernier_num”

Elle permet de déterminer le dernier numéro de dossier qui a été attribué. Cela permet d’éviter les doublons.

4. Fonction “recherche”

Elle permet de faire une recherche, une suppression et une modification de dossier. Nous avons choisi de regrouper ces actions en une seule fonction car dans chaque cas, un parcours de tout un fichier est nécessaire. Cela permet de réduire le programme.

Lors de la suppression d’un dossier patient, le dossier est gardé en mémoire dans un nouveau fichier suivi. Cela permet à l’hôpital d’avoir une traçabilité des dossiers. Les fichiers de suivis sont également séparés avec les consultations dans un fichier “suivi_consultations.txt” et les urgences dans “suivi_urgences.txt”.

5. Fonction “afficher”

Elle permet d’afficher les dossiers selon un critère défini par l’utilisateur.

B. Partie statistique en Python3

Les fichiers txt sont transformés en fichier csv puis lus par le programme python. Nous utilisons les bibliothèques shutil, os et csv.

Pour cette partie statistique, nous utilisons la bibliothèque matplotlib afin d’afficher des représentations graphiques de données de l’hôpital. Nous avons choisi de représenter les données sous diverses formes et ainsi de varier les représentations.

Il est possible d’ajouter de nombreuses autres statistiques, mais nous avons décidé de se limiter à 6 possibilités.