

«Être compétent dans l'usage de L'informatique signifie que l'on sait reconnaître quand émerge un besoin d'information et que l'on est capable de trouver l'information adéquate, ainsi que de l'évaluer et de l'exploiter»

REMERCIEMENTS

Après avoir rendu grâce à DIEU et prier sur le prophète PSL, nous tenons à remercier tous ceux qui ont de près ou de loin ont participé à l'élaboration de ce mémoire :

- Nos parents respectifs
- Notre professeur de C M. Ndiaye
- Notre professeur de développement web M. Fall

INTRODUCTION GÉNÉRALE

Soigner plus : plus de patients, plus vite, plus efficacement, mais avec moins : moins de ressources, de personnel, de budget ; voilà le challenge auquel se trouve confronté aujourd'hui le monde de la santé en général, celui d'une clinique en particulier !

Pour résoudre ce casse-tête, les médecins doivent avoir une approche globale du parcours du patient qui se présente à une clinique, d'y mettre en parallèle les innovations technologiques, afin de proposer de nouveaux modèles de prise en charge qui permettront de solutionner le challenge mentionné en haut.

Avant l'invention du système d'information, on enregistrait toutes les informations manuellement sur des supports en papier ce qui engendrait beaucoup de problèmes tels que la perte du temps. Jusqu'à présent, le système d'informations reste le moyen le plus sûr pour le traitement et la sauvegarde de l'information.

Cette invention a permis d'informatiser les systèmes de données des institutions.

Pour notre projet, les cliniques font partie intégrante des institutions que l'informatique pourra beaucoup aider. Ainsi, nous essayons dans le cadre de ce projet de réaliser une application de gestion des rendez-vous pour les médecins.

Ainsi, l'objectif de notre projet -présenté dans ce Mémoire est la conception et la réalisation d'un système de prise de rendez-vous baptisé **AGENDA ELECTRONIQUE** interactive, fiable, conviviale et facile à s'intégrer dans l'environnement de travail d'une clinique.

Cette application vise essentiellement faciliter le travail des médecins dans la prise d'un rendez-vous. Un médecin peut désormais choisir une plage horaire à sa meilleure convenance et, le cas échéant, annuler un rendez-vous à temps.

CHAPITRE 1 :

Contexte de travail

Introduction

Ce chapitre présente d'une manière générale le contexte de travail et les objectifs de notre projet. Nous allons commencer par une présentation des rôles et les activités de chaque individu, après nous allons présenter le système d'information d'une clinique et ses rôles.

A. Présentation d'une clinique

1) Définition:

La gestion d'une clinique est un emploi complexe. Cependant, les premières années de construction des cliniques, les médecins s'orientaient à un objectif bien déterminé, celui-ci se résumait à comment travailler avec les patients à l'aide de moyens faciles et rapides. Mais malgré tous les efforts des médecins il reste des difficultés, parmi lesquelles: la gestion rendez-vous qui fait perdre beaucoup de temps; ou bien gérer le dossier des patients. A cause de ces difficultés, les médecins préfèrent s'approprier des bénéfices liés à l'informatisation de la gestion de leurs cliniques dans le but de simplifier, sécuriser et minimiser leur travail.

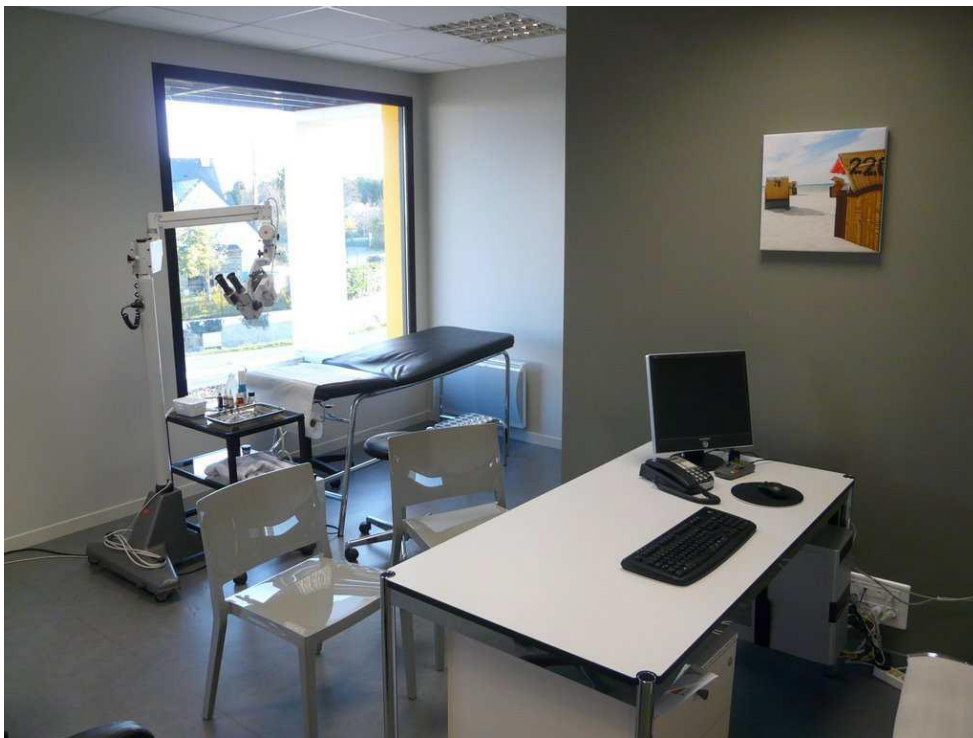









Figure 1.1 : salle de soin

2) Problématiques des cliniques sans les systèmes d'informations

-  Recherche difficile sur les registres.
-  Perte de temps.
-  Détérioration des archives à cause de leur utilisation trop fréquente.
-  Difficulté de stockage.
-  Risque de perte des archives.
-  Mauvaise codification Problèmes dans le calcul des statistiques.
-  Les rendez-vous sont notés dans les agendas.

B. Les systèmes d'informations

Objectif (système informatique) il assure la gestion administrative et médicale et pour cela on a fait introduire l'outil informatique qui gère l'ensemble des rendez-vous sans redondance de l'information et un traitement automatique tout en assurant la sécurité, la fiabilité et la solution qu'apporte notre application. Cela consiste à se débarrasser de plusieurs problèmes tels que la difficulté d'agencer la plage horaire des médecins et celui des rendez-vous.

C. Identification des acteurs

1. Médecin

Son activité principale est de débiter avec des questions simples et tout en montrant la simplicité et une réassurance concernant l'état ou en quelque sorte la maladie en vue de rassurer le patient. En faisant la consultation, le Médecin dispose d'une fiche médicale déjà établie par la secrétaire.



Figure 1.2 : médecin

2. Patient

« Le patient », dérivé de sa racine latine « patiens »: "celui qui souffre", est une "personne recevant une attention médicale ou à qui est prodigué un soin." Le patient est une personne à la recherche d'un meilleur état de santé et ceci par le biais de la médecine et donc des soignants qui l'entoure dans cet objectif. D'après une définition universelle de la santé : « la santé est un état complet de bien-être physique, mental et social, et ne consiste pas seulement en une absence de maladie ou d'infirmité. La possession du meilleur état de santé qu'il est capable d'atteindre constitue l'un des droits fondamentaux de tout être humain, ...».



Figure 1.3 : patient

3. Secrétaire

La profession de secrétaire médicale fait indéniablement le lien entre les **acteurs de santé** et le **patient**. Au carrefour entre les demandes des médecins et celles des patients, la secrétaire doit accueillir chaque personne, avec sérieux, sans se laisser gagner par le stress. La secrétaire médicale doit avoir une expression adaptée à la situation du patient, afin d'être dans l'**empathie**. Elle doit, de ce fait, être digne de confiance et avoir une motivation toute particulière pour cette fonction qui nécessite un sens de la responsabilité et une attention consciencieuse. La secrétaire a un rôle multiple dans la clinique. Pendant l'absence du médecin, elle est seule et doit être en mesure de répondre à tout appel et de recevoir le patient qui se présente au cabinet. Ainsi elle doit écueillier les patients et faire les rendez-vous. En ce sens doit savoir gérer l'accueil et la gestion des rendez-vous.

D. Relation Médecin/Patient

1. Base de communication

Dans tout acte médical, la communication inter active avec le patient est d'une importance fondamentale.

La communication verbale passe par l'intermédiaire de mots, qui sont en général véhiculés par l'expression orale. Dans certaines situations pathologiques la communication nécessitera d'autres moyens, en particulier chez le patient sourd ou chez le patient ayant des troubles des fonctions supérieures.

Les bases de la communication orale avec un patient nécessitent des mécanismes et des temps d'adaptation qui permettront à la communication de s'établir de manière la plus satisfaisante possible dans les deux sens.

Dans un entretien médical, il y a des temps d'écoute où le médecin laisse parler le patient, des temps d'interrogatoires plus dirigistes où le médecin a besoin de préciser des éléments sémiologies indispensables à la bonne prise en charge du patient. Dans tous ces différents temps de la communication orale médecin/patient, il faut absolument que le médecin adapte son langage c'est-à-dire les mots qu'il utilise au niveau socioculturel du patient, et donc en essayant d'utiliser les mots les plus simples du langage lorsque cela est possible. Ces niveaux de compréhension et ces niveaux socioculturels sont différents d'un patient à l'autre et le médecin doit s'adapter en permanence.

2. L'empathie

La relation médecin/malade reste d'un domaine très original et ne ressemble à aucun autre type de relation. Le médecin doit apprendre l'empathie c'est-à-dire la capacité à mettre en place une relation de soutien et de compassion vis-à-vis du patient, sans aller vers des sentiments plus conformes aux relations inter humaines habituelles c'est-à-dire de sympathie ou d'antipathie.

La relation médecin/malade est exigeante et nécessite de plus en plus d'écoute et de capacités pédagogiques des médecins (praticiens).

E. Présentation du projet

Avec les changements des besoins des utilisateurs et l'apparition de nouvelles technologies de stocker les informations et de le trouver quand on veut, de nombreuses personnes cherchent à investir dans le domaine des applications de gestion et chaque entreprise souhaite avoir une bonne gestion de ces systèmes d'informations. La gestion de la clinique est l'ensemble des règles qui facilitent et permettent aux utilisateurs de gérer un grand nombre de tâches afin d'avoir une meilleure organisation du travail.

Après avoir fait critique de l'existant et détecté les anomalies de la procédure actuelle, une approche de solution qui consiste à concevoir et à développer une application qui facilitera les insuffisances et les défaillances énumérées précédemment.

Ce nouveau système disponible vise à simplifier et à accélérer la prise de rendez-vous.

Le système permettra notamment à un médecin d'offrir des plages de rendez-vous à ces patients, il aura accès à un outil sécuritaire, fiable, souple et gratuit qui optimisera la gestion de ses rendez-vous dans la clinique.

Cet outil vise notamment à réduire les rendez-vous manqués et à combler plus efficacement les rendez-vous annulés dans un court laps de temps.

On propose alors de concevoir une nouvelle application permettant la gestion des rendez-vous dans la clinique. En d'autres termes mettre en place un système qui gère les rendez-vous de façon détaillée et rapide pour avoir des informations telles que le traitement etc.

Notre système de prise de rendez-vous baptisé **Agenda Electronique** va permettre la prise de rendez-vous avec un médecin dans la clinique.

F. Gestion des rendez-vous

Avant l'arrivée du système d'information on enregistrerait les rendez-vous dans un agenda.

Un agenda est un outil permettant d'associer des actions à des moments, et d'organiser ainsi son temps.

Il regroupe l'ensemble des jours de l'année, organisés par un découpage qui varie selon les modèles : une page par jour ou par semaine, deux pages par semaine, etc.

Il permet aux personnes d'organiser leur emploi du temps. Il donne à son utilisateur la possibilité de planifier ses activités, de noter son emploi du temps et de rappeler **ses rendez-vous**.

De nos jours l'agenda papier est remplacé par l'Agenda électronique qui peut être partagé pour permettre un travail collaboratif. Les rendez-vous notés dans l'agenda électronique, il contient toutes les informations nécessaires pour la prise d'un rendez-vous.

Il peut être nécessaire d'organiser sa consultation sur rendez-vous si le besoin se fait ressentir et le médecin se doit de les respecter scrupuleusement, le cas échéant, ceci doit être signalé aux patients.

Les gestionnaires de rendez-vous permettent de mieux organiser notre quotidien. Dans cette ère du numérique, disposée d'un agenda électronique est d'un grand intérêt. Surtout dans les secteurs de travail où l'interactivité avec les clients est forte.

Dans le cadre des cliniques, disposée d'un tel outil pourrait être vitale.

Conclusion

Dans ce chapitre on a essayé de faire une étude préalable de notre projet dont le but d'avoir une idée sur la gestion d'une clinique et sur les différentes contraintes de développements.

La gestion des rendez-vous est très difficile en utilisant le papier ou encore l'agenda qui est une méthode archaïque comparée aux outils informatiques, c'est pour cela que la mise en place d'un système de prise de rendez-vous est nécessaire pour faciliter la tâche aux secrétaires et aux médecins.

Avec quoi et comment réaliser cette application ?

CHAPITRE 2:

Conception de l'application

Introduction

Notre projet consiste à mettre en place un agenda électronique. C'est-à-dire, une application permettant au moins d'enregistrer et de voir des rendez-vous. Notre application est une application exclusivement dédiée aux médecins et secrétaires. Ces derniers pourront y échanger afin de synchroniser les heures de visites des patients.

A. Outils de développement

1. Le langage C

Le langage C est un langage de programmation bas niveau (*dans l'ordre des un peu difficile*). La programmation est le faite de communiquer avec la machine et le langage C est le moyen de communication. C'est comme le wolof, le français ou l'anglais entre deux personnes. Le C est un des langages de programmation les plus utilisés, les autres langages plus modernes ont une syntaxe similaire à celle de C et reprennent en parti sa logique.

En contrepartie, la mise au point de programmes en C, surtout s'ils utilisent des structures de données complexes, est plus difficile qu'avec des langages de plus haut niveau. En effet, dans un souci de performance, le langage C impose à l'utilisateur de programmer certains traitements (libération de la mémoire, vérification de la validité des indices sur les tableaux...).

Un peu d'histoire

Le langage C a été inventé au cours des années 1972 dans les laboratoires Bell. Il était développé en même temps que UNIX par Dennis Ritchie et Ken Thompson. En 1978, Kernighan fut le principale hauteur du livre the C « programming decrivant » le langage. Le langage C fut normalisé en 1983. Entre 1994 et 1996 le groupe de travail ISO apporta quelques changements en ajoutant trois fichiers d'entête, un concernant certains nombres macros en rapport avec la norme des caractères et les autres concernant les caractères. En 1999 le C99, une nouvelle évolution à été normalisée par ISO. Avec comme nouveautés les tableaux dynamiques, les nombres complexes... Et puis en 2011 le C11, une nouvelle évolution introduisant le support de la programmation multi-thread, les expressions à types génériques et un meilleur support d'Unicode.

2. La SDL

La SDL est une bibliothèque écrite en C, donc utilisée par les programmeurs en C. Elle est libre et gratuite, on peut réaliser des programmes commerciaux et propriétaires avec. Elle fonction aussi bien sous windows, Max ou linux. C'est une bibliothèque multi-plate-formes et elle est de bas niveau. Nous avons choisi la SDL, parce qu'on avait une main dessus et il fallait encore entrer au fin fond de la bibliothèque en découvrir le contenu. Cela nous permettrait de maîtriser une bibliothèque dédiée au programmeur en C. Le choix n'est pas vraiment intéressant, mais plutôt l'intérêt de la SDL.

La SDL fera de notre application comme toutes celles connu de nos jours. Elle nous fera franchir l'étape de la console. Avant de l'utiliser, nous vous présentons les éléments de la SDL auxquels nous avons fait appel.

3. Le SublimeText3

SublimeText3 est un éditeur multiplateforme qui, sous une apparence déconcertante de simplicité, nous permet de travailler dans un environnement ergonomique et efficace répondant à tous nos besoins de développeur.

Il intègre la plupart des fonctionnalités de base d'un éditeur de texte, dont la coloration syntaxique personnalisable, l'auto complétion, et possède toutes les qualités d'un éditeur à la pointe de nos attentes !

Il est avant tout un éditeur de code, et en ce sens, est un outil qui nous permet d'écrire, de formater, de manipuler des chaînes de caractères, des blocs et des fragments, bref, d'écrire du code. À la différence de certains autres éditeurs de code, SublimeText3 propose un menu principal de recherche qui est autonome.

Si SublimeText3 ne présente qu'un seul espace document, il est possible d'accéder à trois principales palettes : la palette Projet, la palette de navigation et la palette de commande qui, tour à tour, aide à gérer rapidement l'ensemble de nos besoins.

B. Mise en Œuvre

Avant la création de l'application proprement dite, nous avons cherché à définir un système de sauvegarde. Cet élément est le fondement d'un agenda électronique. En effet, imaginons un programme capable de créer, de modifier et de supprimer des rendez-vous. Sans un système de sauvegarde, une fois le programme fermé les données enregistrées disparaissent.

Donc pour pouvoir enregistrer des rendez-vous, il faut moyen d'enregistrement.

1. Système de sauvegarde : les fichiers

En C, les variables n'existent que dans la mémoire vive. Et donc une fois le programme arrêté, ces variables perdent leurs valeurs. Heureusement qu'on peut écrire dans des fichiers et lire ces fichiers qui seront eux stockés dans le disque dur, on ne les perd pas même si le programme est arrêté.

a. GESTION DES FICHIER EN C :

Pour travailler sur des fichiers nous utilisons la bibliothèque stdio. Donc il faut s'assurer d'avoir inclus `<stdio.h>`. Dans cette bibliothèque se trouve des fonctions permettant de gérer les fichiers. Avant de travailler avec les fichiers, il faut leur créer un pointeur. Celle-ci est d'un type particulier, c'est le type **FILE**. La déclaration est de la forme `FILE* fichier = NULL ;`

Les fonctions les plus utilisées sont :

- La fonction `fopen` pour ouvrir un fichier. Le prototype est donné par :
`FILE* fopen(const char* NomDuFichier, const char* ModeOuverture) ;`

Le Nom du fichier pour indiquer le fichier que vous souhaitez ouvrir. Et le mode indique ce que vous souhaitez faire du fichier (lecture seulement, lecture et écriture, écriture seulement, ajout).

- Les fonctions d'écriture `fprintf` et `fwrite` selon le type de fichier.
 - `fwrite` pour les fichiers binaires prototype

```
size_t fwrite(const void *pointeur, size_t size, size_t NombreMembre, FILE* stream);
```
 - `fprintf` sinon. Une fonction au mode de fonctionnement similaire à celui de `printf`. Assez facile à utiliser.
- Les fonctions de lecture `fscanf` et `fread` selon le type de fichier :
 - `fread` pour le binaire, prototype :

```
size_t fread(const void *pointeur, size_t size, size_t NombreMembre, FILE* stream);
```
 - `fscanf` sinon. Fonctionnement similaire à `scanf`.
- La fonction `remove` permet de supprimer un fichier. Syntaxe `remove(Nom_Du_Fichier);`
- Et la fonction de fermeture `fclose` :

```
int fclose(FILE* NomFichier);
```

b. LES FICHIERS BINAIRES :

Un fichier binaire, en informatique, est un fichier qui n'est pas un fichier texte. Ils permettent aussi de stocker des informations sous forme de bits. Ci-dessus, nous avons énuméré les fonctions permettant de lire et d'écrire dans des fichiers binaires. Nous comptons, en faites, utiliser les fichiers binaire pour enregistrer nos rendez-vous. Parce qu'il est plus facile de stocker et de récupérer des structures dans des fichiers binaires.

c. DÉFINITION DU SYSTÈME SAUVEGARDE :

- ✚ Chaque médecin disposant d'un dossier se trouvant au même endroit que l'exécutable.
- ✚ Les rendez-vous sont des structures. On les stocke dans des fichiers binaires.
- ✚ Ces fichiers sont distribués dans les dossiers des médecins où on les enregistre.
- ✚ Chaque dossier contiendra les rendez-vous du médecin correspondant.
- ✚ Pour des éventuelles modifications ou suppressions, on récupère les fichiers binaires soit pour modifier le contenu ou supprimer le fichier.

Pour mettre en pratique ce système, nous avons réalisé une application en console.

2. Illustration en mode console

L'objectif de cette application en console est de mettre en évidence de manière concrète le système de sauvegarde défini ci-dessus.

a. FONCTIONS CRÉÉES :

L'application sert à enregistrer des rendez-vous sous forme de fichiers binaires. Elle permet de les modifier, de les supprimer et de les afficher. Pour chacune de ces tâches nous avons créé un fonction d'exécution :

- ❖ Fonction d'accueil :
Elle oriente l'utilisateur en lui présentant un menu composé de cinq choix : enregistrement, modification, suppression et modification. Selon le choix, on fait appelle au fonctions suivantes.
- ❖ La fonction d'enregistrement
Celle là ouvre un menu demandant le médecin pour lequel on crée un rendez-vous. Une fois ce dernier sélectionné, elle donne la main à l'utilisateur pour renseigner les identifiants du patient et son rendez-vous. La termine par créer un fichier binaire qu'elle enregistrera dans le dossier du médecin correspondant. Ce fichier contiendra les identifiants et le rendez-vous du patient.
Comme la console n'est que pour une mise en évidence, nous avons délibérément choisi un nombre fini de rendez-vous pour chaque médecin. C'est pour cela, que chaque rendez-vous créé est accompagné d'un numéro.
- ❖ La fonction de modification et la fonction de suppression:
Comme la précédente, il faudra indiquer le médecin. Cela permet à aller chercher le fichier binaire au bon dossier. Ces fonctions demandent à l'utilisateur d'indiquer le numéro du patient permettant de sélectionner le bon fichier. Soit on le lis et modifie le contenu ou on le supprime.
- ❖ La fonction d'affichage :
Toujours il faudra indiquer le médecin. La fonction d'affichage ouvrir, ainsi, les fichiers contenu dans ce dossier un à un en mode lecture.

b. DEMONSTRATION :

Création de dossier :

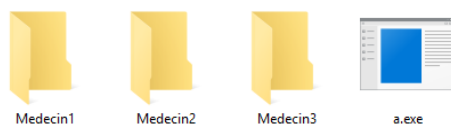


Figure 2.1 : Ces dossiers sont placer au même endroit que l'exécutable. Et il faut les créer manuellement.

Menu

```

-----GESTIONNAIRE DE RENDEZ-VOUS-----
1.Enregistrer un rv      2.Modifier un rv      3.Supprimer un rv
4. Voir les rv

```

Création de rendez-vous

Nous le ferons pour deux médecins (1 et 3), afin de voir le partage.

```

1. Medecin 1      2. Medecin 2      3. Medecin 3

```

```

-----Renseigner les champs-----
Nom : Sabaly
Prenom : Thierno
age : 20
sexe(M/F) : M
-----Son rendez-vous-----
Jours : Lundi
Heure : 12
Minutes : 30

```

Le rendez-vous pour le médecin 1.

```

-----Renseigner les champs-----
Nom : Thiam
Prenom : Tidiane
age : 20
sexe(M/F) : M
-----Son rendez-vous-----
Jours : Mercredi
Heure : 15
Minutes : 0
5 => pour quitter Quitter, autre chiffre sinon

```

Le rendez-vous du médecin 3.

Les dossiers après cette action :

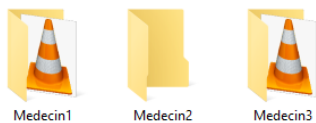


Figure 2.2 : dossier console, après création de rendez-vous

Affichage

```
1. Medecin 1    2. Medecin 2    3. Medecin 3
```

Toujours on indique le médecin

```

patient n_1
Patient : Nom = Sabaly    Prenom = Thierno    Sexe = M    age : 20ans
Rendez-vous : Lundi a 12h30mn
-----
5 => pour quitter Quitter, autre chiffre sinon

```

médecin1

```

patient n_1
Patient : Nom = Thiam    Prenom = Tidiane    Sexe = M    age : 20ans
Rendez-vous : Mercredi a 15h0mn
-----
5 => pour quitter Quitter, autre chiffre sinon

```

médecin3

Modification

```
1. Medecin 1    2. Medecin 2    3. Medecin 3
```



```

Indiquer le numero du patient : 1
(1) Modifier le rv
(2) Remplacer le client
1
Jours : Jeudi
Donner l'heure : 16
Minutes: 0
5 => pour quitter Quitter, autre chiffre sinon

```

On modifie ici, le rendez-vous du patient pour le

médecin 1

Affichage :

```

patient n_1
Patient : Nom = Sabaly   Prenom = Thierno       Sexe = M       age : 20ans
Rendez-vous : Jeudi a 16h0mn
-----
5 => pour quitter Quitter, autre chiffre sinon

```

Suppression :

```

1. Medecin 1    2. Medecin 2    3. Medecin 3

```

Nous allons supprimer le rendez-vous du médecin 1.

```

Vous voulez supprimer un rv : (1) oui ou (2) non : 1
Donner le numero du patient : 1
5 => pour quitter Quitter, autre chiffre sinon

```

Résultat :

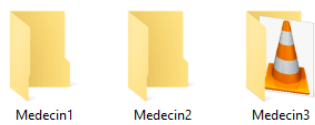


Figure 2.3 : dossier console, après suppression de rendez-vous

Cette application en console nous prouve l'efficacité des fichiers dans la sauvegarde et la restitution des rendez-vous. Les fichiers sont accessibles tout le temps. Ce mode de fonctionnement est le modèle de base de la réalisation en fenêtre de l'application. Les codes sources sont très différents, mais en arrière plan c'est le même système.

3. L'interface graphique

Nous avons trouvons de fonctionnement fiable et durable avec les fichiers. Il ne nous reste plus qu'une bonne présentation de l'application afin qu'elle soit attirante, plus communicante, plus fiable... Il ne reste plus qu'une intégration à la SDL.

Le système de sauvegarde est le même. Par contre, ici il est plus complexe.

Redéfinition du système :

Il faut :

- créer un dossier, au même endroit que l'exécutable, appelé "**Medecins**"
- Dans ce dernier, on crée un dossier pour chaque médecin de la clinique.
- Dans le dossier de chaque médecin, on crée deux sous-dossiers. Un appelé "**disponibilités**" et l'autre "**RendezVous**".
- De même, on crée un dossier "Secrétaires" et chaque secrétaire on lui crée un sous-dossier dedans en sont nom.

Ceci doit être fait avant toute utilisation de l'application.

Ainsi le mode de fonctionnement est le suivant :

- ✚ Les disponibilités seront des fichiers binaires
- ✚ Des fichiers qui se trouveront dans le dossier du médecin qui les aura indiqués. Ils seront plus précisément dans le sous-dossier "**disponibilités**".
- ✚ Le nombre de disponibilité sera compté et stocké dans un fichier binaire stocké dans le dossier du médecin correspondant.
- ✚ Les rendez-vous seront des fichiers binaires.
- ✚ Ils seront eux stockés dans le sous-dossier "**Rendezvous**" du médecin correspondant.
- ✚ Le nombre de rendez-vous sera aussi compté et stocké dans un fichier binaire. Ce dernier sera conservé dans le dossier du médecin correspondant.

a. LES ÉLÉMENTS DE LA SDL QUE NOUS AVONS UTILISÉ

La SDL nous a permis de créer une fenêtre dans laquelle nous insérons du texte et des images. Elle nous permet aussi de créer des surfaces que nous pouvons colorier, utilisé à nos fins. Avec la SDL :

- *On gère les déplacements et les clics de la souris de la souris*
Un déplacement de la souris est stocké dans un événement de type **SDL_MOUSEMOTION** et les clics un événement de type **SDL_MOUSEBUTTONDOWN**.
Pour chacune des fonctions que nous avons créées dans notre programme, nous leur envoyant deux variables pour la gestion de la souris. Une variable pour les déplacements de la souris permettant de repérer les survols des boutons et l'autre repère les clics opérant dans cette fonction.
- *La gestion du clavier*

C'est un événement du type **SDL_KEYDOWN** qui repère un événement du clavier. C'est-à-dire, quand on appuie sur une touche du clavier. Nous permettant, ainsi de récupérer les touches appuyées. Utile pour la saisie.

b. LES FONCTIONS CRÉÉES

Ici nous avons créé un bon nombre de fonction. Nous ne présentons que les plus importantes.

❖ Les fonctions de création de compte

Il y en a deux une pour les secrétaires et une autre pour les médecins. Ce sont les deux autorisés à utiliser l'application, ils doivent s'assurer d'avoir un dossier en leur nom. Les fonctions de création de compte présentent trois champs qui sont : le nom, le mot de passe et une confirmation du mot de passe.

Elles regardent d'abord si votre dossier existe et que les mots de passe entrés sont identiques. Si oui, elles créent un fichier binaire, dans le dossier du médecin correspondant, stockant le nom et mot de passe.

Prototypes :

```
void Creation_Compte_Med(SDL_Surface* fenetre, identifiant id, Bout_Creer_Compte
CC_App, char masqueur1[50], char masq_check[50],char masqueur2[50], char
confirmation[50]) ;
```

```
void Creation_Compte_Sec(SDL_Surface* fenetre, Bout_Compte_Sec Bout_Appuye,
identifiant id, char masqueur1[50],char masqueur2[50], char verifie[50]) ;
```

❖ Les fonctions de connexion

Là aussi nous en avons deux. Pour se connecter, on renseigne deux champs le nom et le mot de passe. Une fois validé, la fonction ouvre dans le dossier portant ce nom, le fichier binaire stockant le nom et le mot de passe. Si ce fichier existe, on récupère les informations s'y trouvant pour comparer avec celles saisies. Sinon, c'est parce que le dossier n'existe pas.

Prototypes des deux fonctions :

Médecins :

```
void connecte(SDL_Surface* fenetre, identifiant* id, Bout_ID ID_App, char masqueur1[50],
int *connected, char confirmation[50],char Med_Conn_Nom[50]) ;
```

Secrétaire :

```
void Connexion_Sec(SDL_Surface* fenetre, Bout_Connect_Sec Bout_Appuye, identifiant id,
char masqueur1[50], int *connected) ;
```

❖ La fonction permettant de créer des disponibilités

Elle n'existe que pour les médecins. A travers elle le médecin peut indiquer la date (jours-mois-année) et l'heure à laquelle il est libre pour un rendez-vous. Une fois les champs renseignés, elle ouvre dans le dossier du médecin correspondant le sous-dossier **disponibilités**. Dans ce dernier, elle crée un fichier binaire stockant la date et l'heure. En plus, si une date est déjà prise le fichier n'est pas créé car déjà existant.

Prototype :

```
void Entrer_Dispo(SDL_Surface* fenetre, char dossier[50],Rendez_Vous Cr_Rv,
Dispo_Creation DC_App, int *nombre_Dispo,char Dossier_En_Cours[50]) ;
```

❖ La fonction permettant de créer les rendez-vous

Cette partie est dédiée à la secrétaire. Elle renseigne d'abord le nom du médecin (comme dans la console d'ailleurs). Une fois le nom renseigné, la fonction ouvre le dossier "Medecins" à la recherche d'un sous-dossier portant ce nom. Dans le cas où elle en trouve, elle donne la main à la secrétaire pour renseigner le nom, prénom, sexe et l'âge. En même temps, elle ouvre le dossier contenant les disponibilités de ce médecin pour permettre à la secrétaire de faire un choix. Lorsqu'elle valide, on crée un fichier binaire dans le sous-dossier **RendezVous** du médecin correspondant.

Prototype :

```
void Creer_Patient(SDL_Surface *fenetre, Bout_CreerPatient_Sec Bout_Appuye, char
Medecin[50], Patient patient, int *compteur,int *Nbr_Rv, int *Nbr_Dispo) ;
```

❖ Les fonctions d'affichages

Elle permette aux médecins de voir leurs disponibilités et leurs rendez-vous. Elles sont faciles à concevoir car elles ne font que lire les fichiers se trouvant dans les sous-dossiers **disponibilités** et **RendezVous** du médecin connecté.

Prototypes :

```
void BlitRv(SDL_Surface* fenetre, int Nbr_Rv, char Dossier[50]) ;
```

```
void BlitDispo(SDL_Surface* fenetre, char Dossier[50], int nombre_Dispo) ;
```

c. DÉMONSTRATION

En pratique voyons ce qui se passe en arrière plan pour chacune de nos actions.

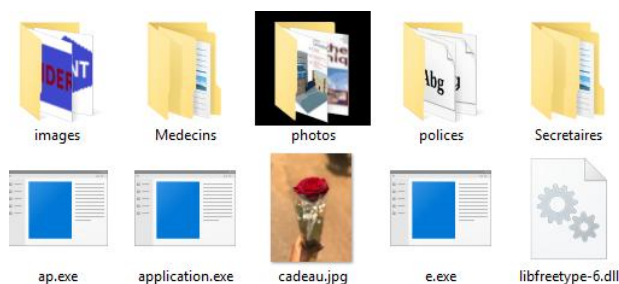


Figure 2. 4 : Voici un dossier "Medecins" avec l'exécutable.

Créons dedans un dossier du nom de "sall".

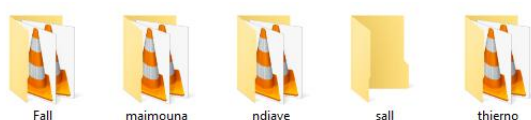


Figure 2.5 : création du dossier sall

Dans le dossier "sall", créer les deux sous dossiers "disponibilités" et 'RendezVous'

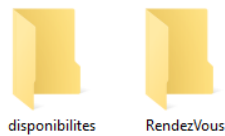


Figure 2.6 : Les sous-dossiers de sall

Sall est maintenant prêt à utiliser l'application.

- **Création de compte :**
Un compte ici correspond à une activation de dossier. Actuellement, M. Sall a un dossier mais il ne pourra l'utiliser sans créer de compte.

Voyons le dossier de sall après la création de compte.

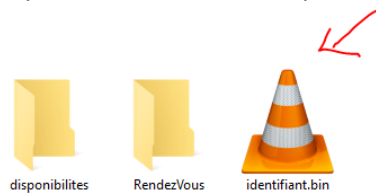


Figure 2.7 : dossier de sall, après création de compte

- **Connexion :** Revoyons le dossier, une fois sall connecté.

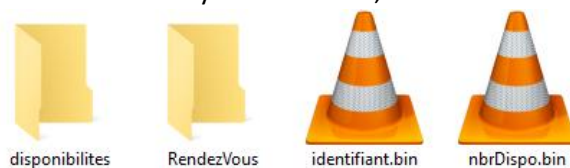


Figure 2. 8 : dossier de sall, après connexion

Un fichier de stockage du nombre de disponibilité est généré.

- Désormais, M. Sall a accès à son dossier. Il peut créer ses disponibilités (initialement vide). Créons en deux, et revoyons son dossier



Figure 2. 9 : dossier de sall, après création de disponibilité.

Des disponibilités sont créées.

- **Les rendez-vous**
C'est au secrétariat que se créent les rendez-vous. On va utiliser un compte que j'ai déjà créé afin de créer un rendez-vous pour sall. Voyons le dossier de sall.



Figure2.10 : dossier de sall après création d'un rendez-vous.

Un fichier gérant le nombre de Rendez-vous est généré plus le rendez-vous.

Conclusion

Nous avons réalisé une application desktop à base du langage C. Les fichiers nous ont permis de créer un système de sauvegarde. Nous avons choisi de créer notre base de données avec les fichiers, pour mieux assimiler la notion « gestion des fichiers en C » à travers ce projet. Pour le graphisme, nous avons fait appel à la SDL. Une bibliothèque développée en C, donc cela nous permet de maîtriser une bibliothèque dédiée au langage C. Enfin, nous avons utilisé SiblmeTexte3 comme éditeur.

Au début, c'était une application console pour tester le système de sauvegarde. Ce dernier a été maintenu lors de la réalisation de l'application en mode fenêtre. Les captures des besoins de cette application feront l'objet du chapitre suivant.

CHAPITRE 3:

Présentation de l'agenda électronique

Introduction

Dans ce chapitre nous allons présenter la conception de notre système baptisé **AGENDA ÉLECTRONIQUE**, nous présentons l'implémentation de l'application gestion de rendez-vous, l'architecture utilisée. Ensuite nous présentons les écrans les plus importants de notre application.

A. Description des interactions

1. Authentification:

Processus par lequel le système s'assure de l'identité de l'utilisateur, il doit s'authentifier par un login et un mot de passe. Dans l'Agenda électronique les deux utilisateurs sont les médecins et la secrétaire qui auront chacun un identifiant et un mot de passe. L'authentification se fait selon le nombre de fois qu'il ouvrira son compte.

La prochaine connexion dans l'application sera prise en compte avec son identifiant que ce soit médecins ou secrétaire.

2. Réserver un RDV :

Pour réserver une consultation avec un médecin de la clinique, vous aurez besoin de parler à la réceptionniste de la clinique qui se chargera de vous renseigner sur les plages horaires des médecins c'est-à-dire les heures auxquelles les médecins sont disponibles pour prendre un RDV avec un médecin.

Les rendez-vous sont enregistrés dans les fichiers binaires que le médecin pourra ensuite voir dans son compte.

3. Modifier un RDV :

Si le patient ne peut pas se présenter à un RDV, il a le droit également de faire des modifications, en avertissant la réceptionniste qui se chargera de changer le rendez-vous du patient, en parallèle elle vérifiera si les modifications sont possibles (date, heure) c'est juste après que le patient aura le choix de faire ou non sa modification.

Il peut aussi modifier ses informations personnelles (nom, prénom) qui seront prises en compte par l'application.

4. Annuler un RDV :

Le patient a un empêchement. Il ne peut pas se rendre à un rendez-vous, il a le droit d'annuler son rendez-vous avant sa date prévue. Il doit appeler la réceptionniste pour confirmer son annulation. Désormais la réceptionniste aura la possibilité de recontacter les clients qui ont voulu avoir cette date maintenant libérée. Le premier avoir répondu à son contact aura le rendez-vous validé.

B. Les interfaces de l'application

Dans notre application on a deux utilisateurs par défaut. Par conséquent, nous avons prévu un login et un mot de passe (compte) pour chacun d'eux à savoir les médecins et la réceptionniste (qui sont déjà ajoutées dans les dossiers au cours de l'installation de l'application).

1) Accueil :

À la lance de notre application, une fenêtre s'affiche à l'écran, celle de l'accueil avec une zone info, et une bref présentation en quelque sorte de la clinique.



Figure 3.1 : Accueil


2) Espace secrétaire :

Espace secrétaire : contient : **Zone d'Alerte**, **Connexion** (se connecter, se déconnecter), **Créer un compte**, **Créer un RDV**.

Pour que la secrétaire puisse se connecter ou créer un compte, il faut qu'il dispose d'un dossier dans le réseau local de la clinique. Ensuite il pourra accéder au compte des médecins pour voir les plages horaires de ces derniers.



Figure 3.2 : Espace secrétaire

-  A l'arrivée d'un nouveau patient la secrétaire remplit une nouvelle fiche. La gestion des rendez-vous est une tâche essentielle de la secrétaire, celle-ci vérifie la disponibilité de la date demandée et par la suite elle ajoute un rendez-vous en saisissant les renseignements nécessaires.

Creation d'un patient

Indiquez ci-dessous le nom du medecin pour lequel vous creez le rendez-vous

ndiaye| **SUIVANT**

Information sur le patient

Nom Prenom

Age Femme ☐ Homme ☒

Traitement

La date du rendez-vous

Le 11/10/2017 a 11h 57min

VALIDER

Figure 3.3 : création de patient

3) Espace Médecin :

Espace médecin contient les deux fonctionnalités qui sont : **Mes disponibilités, Mes rendez-vous.**

Le médecin doit disposer au préalable d'un dossier pour pouvoir créer un compte.

Agenda Electronique

Accueil EspaceSecrétaire **Espace_Medecin** A propos

Mes Disponibilités Mes Rendez-Vous

Vous n'êtes pas connecté

Après identification, chaque médecin pourra renseigner ses disponibilités en indiquant seulement la date (le jour et l'heure). Le secrétaire se chargera de créer les rendez-vous pour chaque date indiquée. Avant tous cela, il faut d'abord d'un compte d'utilisateur.

Connectez-vous
Créer un compte

Se déconnecter

DATE
Thu Oct 11 13:06:55 2018
Assurez-vous que votre PC soit toujours à jours, pour la bonne gestion du temps

BIENVENUE Dans l'espace des médecins
Voici quelques éléments à savoir :
La création d'un compte est équivalent à la création d'un dossier.
Pour que chaque médecin n'accède qu'à son dossier, il doit lui créer un compte d'utilisateur.
Accueille Médecin

ICI

une clinique est un centre hospitalier
Souvent spécialisée en un domaine bien précis il y'a qui sont privées et d'autres sont publiques.

Figure 3.4 : Espace médecin

Une fois crée il pourra ensuite se connecter dans son compte et remplir ses plages horaires ou encore voir ses rendez-vous du jour et les rendez-vous à venir.

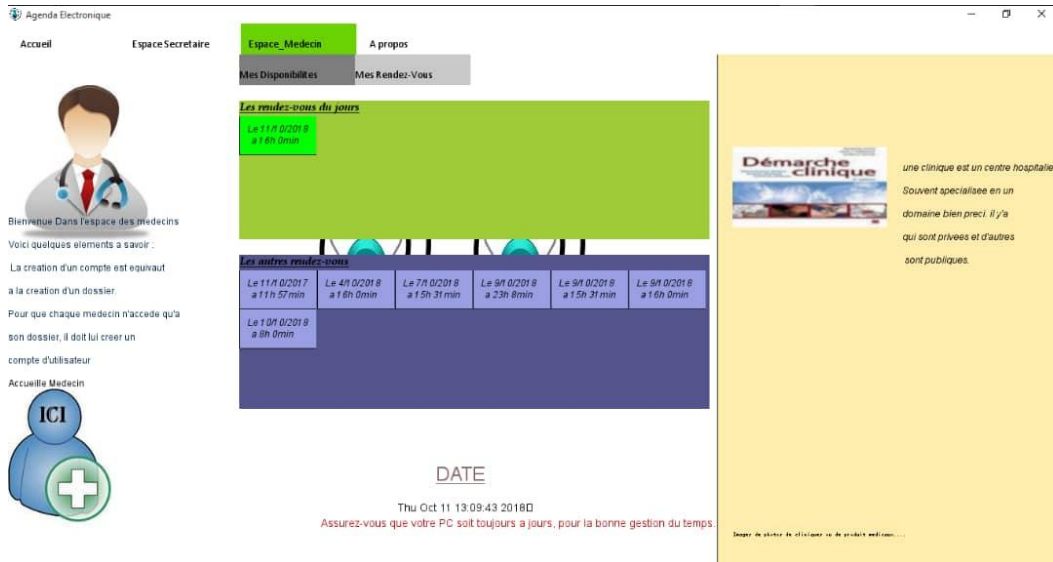


Figure 3.5 : rendez-vous

Conclusion:

A la fin Dans ce chapitre, nous présentés le côté implémentation de notre projet, et les différents outils du développement de notre application ainsi que ses interfaces essentielles.

CONCLUSION GÉNÉRALE

Gérer efficacement les rendez-vous d'une clinique est un défi au quotidien car cela requiert un équilibre permanent entre la gestion du personnel, et le maintien de la satisfaction des patients. Voilà pourquoi plusieurs cliniques optent pour l'intégration des technologies pour optimiser leur fonctionnement.

Nous avons remarqué durant ce travail, que la gestion des rendez-vous représente une activité très importante au sein d'une clinique, et que sa facilité de gestion est un facteur important qui doit exister dans toute clinique. Notre travail avait comme but d'améliorer la méthode actuelle de gestion, commençant par une facilité de la prise des rendez-vous.

Dans ce mémoire de projet de fin d'étude, nous avons regroupé toutes les étapes de la conception et la réalisation de notre solution informatique sur la gestion de rendez-vous d'une clinique appelée Agenda électronique, ainsi construit qui fonctionne correctement et répond aux besoins énoncés, mais qui peut notamment être amélioré.

Donc nous avons eu l'opportunité d'approfondir nos connaissances que ce soit au plan scientifique ou personnel mais aussi ceci nous a permis d'avoir une bonne expérience et amélioration de notre connaissance concernant le domaine de la programmation.

Comme les autres expériences au niveau de l'application des concepts de langages, c'est normal de ne pas pouvoir éviter certains problèmes et difficultés au niveau de la conception et la programmation. Cependant nous avons essayé de dégager les solutions les mieux adaptés à nos objectifs car, en informatique y'a pas de solution absolue.