CPNV

Documentation

Projet Prog

Système de gestion de centre médical

MARTIN-PEÑALOSA Beatriz

MAYOR Damien

20.01.2022

Table des matières

[1 Système de gestion d’un centre médical 2](#_Toc125103692)

[1.1 Mise en situation 2](#_Toc125103693)

[1.1.1 Cahier des charges 2](#_Toc125103694)

[1.1.2 Descriptif 2](#_Toc125103695)

[1.2 Cas d’utilisation et scénarios 3](#_Toc125103696)

[1.3 Planification des tâches 3](#_Toc125103697)

[1.4 Tests effectués 4](#_Toc125103698)

[1.5 Rétrospective 4](#_Toc125103699)

[1.5.1 Problèmes rencontrées 4](#_Toc125103700)

[1.5.2 Ce qui n’a pu être abouti 5](#_Toc125103701)

[1.5.3 Apprentissages 6](#_Toc125103702)

[1.6 Marche à suivre 7](#_Toc125103703)

[1.6.1 Mise en place de la base de données dans HeidiSQL 7](#_Toc125103704)

[1.6.2 Utilisation de l’application 7](#_Toc125103705)

[1.7 Conclusion 10](#_Toc125103706)

[1.8 **Annexe 1** : Modèles de données 11](#_Toc125103707)

[1.8.1 MCD 11](#_Toc125103708)

[1.8.2 MLD 12](#_Toc125103709)

[1.9 **Annexe 2** : Organigramme 13](#_Toc125103710)

[1.10 **Annexe** **3** : Maquettes 13](#_Toc125103711)

[1.11 **Annexe** **4** : Moodboard 16](#_Toc125103712)

[1.11.1 Apparence formulaires 16](#_Toc125103713)

[1.12 **Annexe** **5** : Journal de travail 17](#_Toc125103714)

[1.13 **Annexe 6** : Journal de bord 19](#_Toc125103715)

# Système de gestion d’un centre médical

## Mise en situation

### Cahier des charges

Cette application en C# a pour but de gérer un centre médical.

Elle va donc nous permettre de :

* S’identifier pour se connecter et avoir accès aux données
* Ajouter de nouveaux patients
* Voir la liste des patients répertoriés dans la base de données
* Voir leurs anciennes consultations
* Modifier leurs données
* Supprimer un patient de la liste
* Créer une nouvelle consultation pour un patient sélectionné
* Modifier une consultation

### Descriptif

Un patient est assuré par une seule assurance. On nécessite uniquement du nom de l’assurance.

Le patient va être caractérisé par les données suivantes : son nom et prénom, sa date de naissance, son numéro AVS, son numéro de téléphone et son adresse e-mail, son adresse de résidence (rue, localité, NPA). On inscrit également depuis quand il est patient dans le centre.

Un patient va avoir une consultation à une date précise, pour une raison donnée avec un soignant spécifique qui va entrer une description de la consultation. La consultation comporte un numéro unique.

Un soignant va être défini par son nom et prénom, son acronyme, sa spécialisation et son numéro de téléphone. Il peut traiter plusieurs patients, ou zéro s'il est nouveau dans le centre. On sait quelles consultations sont les siennes.

Une consultation peut aboutir à aucun diagnostic et traitement (si tout va bien pour le patient) comme à plusieurs. Un diagnostic comporte un numéro unique et un traitement.

Chaque soignant à sa propre salle attitrée. Il existe différents types de salles (consultation, radiographie, laboratoire, scanner, salle de soins). Certaines salles n'ont pas de soignant attitré car il s'agit de salles d'examens complémentaires qui sont occupées à des moments très spécifiques (scanner, radiographie, une des salles de soins).



## Cas d’utilisation et scénarios

L’utilisateur du logiciel :

* Va s’identifier avec un identifiant et mot de passe pour accéder à la page d’accueil et avoir accès aux données et fonctionnalités.
* Sur la page d’accueil, va pouvoir se déconnecter pour revenir à la page de connexion.
* Va pouvoir ajouter un nouveau patient à la base de donnée grâce à un formulaire en cliquant sur Ajouter un patient.
* Va pouvoir avoir accès à la liste des patients répertoriés dans la base de données en cliquant sur Rechercher.
* Dans Rechercher, va pouvoir sélectionner un patient existant pour modifier ses données personnelles.
* Dans Rechercher, va pouvoir sélectionner un patient existant pour le supprimer de la base de données.
* Dans Rechercher, va pouvoir sélectionner un patient existant pour ouvrir son dossier de patient et voir ses consultations passées.
* Dans la liste des consultations d’un patient, va pouvoir sélectionner une consultation existante pour la modifier.
* Dans la liste des consultations d’un patient, va pouvoir créer une nouvelle consultation.

## Planification des tâches

|  |  |  |
| --- | --- | --- |
| **QUoi** | **quand** | **qui** |
| Choix du sujet | Semaine 1 | Damien, Beatriz |
| MCD | Semaine 1-2 | Damien, Beatriz |
| MLD | Semaine 1-2 | Beatriz |
| Diagramme | Semaine 2-3 | Damien |
| Maquette | Semaine 2-4 | Damien |
| Organigramme | Semaine 4 | Damien, Beatriz |
| Script DB | Semaine 2-4 | Beatriz |
| Connexion application avec DB | Semaine 3-5 | Beatriz |
| Fichier JSON pour identifiants connexion à la DB | Semaine 5 | Beatriz |
| Partie patients (C# + formulaires + queries SQL) | Semaine 3-8 | Beatriz |
| Partie consultations (C# + formulaires + queries SQL) | Semaine 5-8 | Damien |
| Design, mise en forme | Semaine 7-8 | Damien, Beatriz |
| Tests | Semaine 6-8 | Damien, Beatriz |
| Documentation | Semaine 1-8 | Damien, Beatriz |

## Tests effectués

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **test** | **resultat** | **qui** |
| **Connexion à l’application** | | | |
| **21.12.2022** | Identifiant et mot de passe corrects | **OK** | Beatriz |
| Identifiant erroné et mot de passe correct | **OK** |
| Identifiant correct et mot de passe erroné | **OK** |
| Identifiant manquant et mot de passe correct | **OK** |
| Identifiant correct et mot de passe manquant | **OK** |
| **Ajout d’un nouveau patient** | | | |
| **11.01.2023** | Informations du patient complètes et correctes | **OK** | Beatriz |
| Information du patient manquantes | **OK** |
| Informations du patient devant être unique dupliquées | **NOT OK**  **Exception** |
| **17.01.2023** | Informations du patient devant être unique dupliquées | **OK** | Beatriz |
| **Modification d’un patient existant** | | | |
| **17.01.2023** | Informations du patient complètes et correctes | **OK** | Beatriz |
| Informations du patient non complètes (effacées) | **NOT OK**  **Exception** |
| Informations du patient devant être unique dupliquées | **NOT OK**  **Exception** |
| Informations du patient non complètes (effacées) | **OK** |
| Informations du patient devant être unique dupliquées | **OK** |

## Rétrospective

### Problèmes rencontrées

Malheureusement, ce projet n’a pu être finalisé comme nous le souhaitions. Divers obstacles ce sont mis en travers de notre chemin et ont ralenti notre progression.

Tout d’abord, il est important de mentionner que le projet en soi était trop grand par rapport au temps que nous avions à disposition. De plus, nos compétences en matière de programmation en C# comportent énormément de lacunes. Ce qui fait que manque de temps plus manque de compétences sont très liés au final.

La mise en place du projet nous a pris plus de temps que prévu, car manque d’expérience dans le domaine, nous n’avons pas su nous centrer correctement et avons perdu du temps avec de nombreux détails, notamment avec la réalisation du MCD (beaucoup de changements apportés durant les premières semaines) ou encore avec la réalisation trop tardive d’un diagramme de flux pour nous aider à visualiser notre application de meilleure façon.

Au moment de coder, de nouvelles notions sont apparues. En effet, le DataGridView utilisé pour afficher la liste des patients n’a jamais été abordé en classe. Ceci a entraîné une perte de temps considérable dans la compréhension du fonctionnement et dans la mise en place de celui-ci. A ce sujet, nous tenons à remercier Madame Frédérique Andolfatto pour son aide précieuse apportée.

La récupération des informations depuis la base de données pour ensuite les afficher dans ce DataGridView a donc été sinueux. Et venu ensuite le moment de pouvoir sélectionné un patient dans ce DataGridView pour ouvrir un nouveau formulaire récupérant les données personnelles du patient pour modification. A nouveau, la majorité du temps a été passé dans les forums pour trouver des explications et des exemples correspondant le mieux à notre situation. Ceci n’a pas été chose facile au vu de la complexité de la programmation orienté objet et nos faibles connaissances.

De grosses disparités dans nos capacités à coder, différence connue et discutée avant même le début du module se sont montrer plus impactantes qu’anticipé.

Nous avons rencontré un certain nombre de difficultés avec Github tant lors de fusion de branches ou de simple push menant à des pertes de données à plusieurs reprises et à une perte de confiance dans cette plateforme.

N’ayant jamais effectué de projet de telle ampleur, gérer toutes les pages de notre code, se rappeler que telle ou telle fonction interagit avec tel formulaire ou non, a pu créer de la confusion et donc, encore une fois, freiner notre avancement.

Pour finir, l’orienté objet a également été source de problèmes lors de la réalisation du projet, car la compréhension détaillée de comment fonctionnent une classe et un objet n’était clairement pas acquise et nous a donc ralenti.

### Ce qui n’a pu être abouti

Comme le projet n’a pu être finalisé, nous allons tout de même mentionner ce qui n’a pas été implémenté, mais que nous aurions voulu intégrer dans notre application.

Tout d’abord, lorsque l’on ajoute un nouveau patient ou modifions les données d’un patient existant, il n’y a pas de vérification de format. Si le temps nous l’avait permis, avec des RegEx (Regular Expressions) nous aurions pu vérifier chaque format désiré dans chaque champ.

Par exemple, pour le nom et prénom, uniquement des lettres auraient été admises ("^[a-zA-Z]+$"). Ou encore pour le code postal, seulement 4 chiffres auraient validés le champ ("^[0-9]{4}$").

Un autre détail qui aurait été modifié par la suite est le login à l’application. En effet nous avons uniquement pu faire un login très générique d’un admin, mais le but aurait été d’avoir diverses adresses email faisant partie du centre médical avec un mot de passe. Le plus gros du travail avait été déjà fait, mais au moment de faire des choix, la priorité s’est porté sur d’autres fonctionnalités.

Dans le DataGridView, la taille des colonnes aurait été modifiée pour que la colonne du code postal soit moins large par exemple.

De plus le bouton Supprimer aurait été implémenté pour supprimer un patient de la liste et donc de la base de données. Bien entendu, dans la réalité et pour des raisons légales, nous ne supprimerions pas un patient, même si celui-ci ne vient plus ou est décédé.

Le dossier patient n’affiche pas les données des patients contenues dans la base de donnée, ni leurs consultations. Par conséquent il n’est possible de rentrer dans une consultation afin d’en voir le détail, de les modifier ou même de les supprimer.

Concernant Le formulaire d’ajout de consultation à cause du manque de temps, l’ajout des données à la base de donnée n’a pas encore été implémenté.

L’apparence de l’application aurait été retravaillée (boutons crées sur Photoshop, icône insérée à divers endroits), mais faute de temps, nous avons dû nous contenter d’une apparence très simple et basique. Une apparence qui reste tout de même cohérente, mais qui ne correspond pas nos attentes.

Nous aurions également désiré mettre une page de chargement au lancement de l’application, avec le logo et un cercle autour montrant le pourcentage de chargement, mais la priorité a été mise, encore une fois, sur le code et nous ne nous sommes donc pas penché sur cette fonctionnalité méconnue de notre humble savoir.

### Apprentissages

Bien entendu, nos réussites et nos échecs ne sont jamais en vain et ceux-ci nous ont permis d’apprendre de nombreuses choses au fil du projet.

Tout d’abord, ne pas avoir les yeux plus gros que le ventre. En effet, il est important de faire une introspection et se rendre compte où sont nos limites. Vouloir créer un projet très complet c’est bien beau, mais si les compétences ne sont pas présentes, le projet sera voué à l’échec, ce qui apporte de la frustration.

Mais c’est grâce à ce genre d’erreur que nous pouvons mieux faire la prochaine fois. Il faut tenir en rigueur que c’est la première fois que nous nous retrouvons ainsi lâchés dans la nature.

Au niveau de l’apprentissage de programmation orienté objet en C#, une meilleure compréhension de ce qu’est une classe et de ce qu’est un objet a été acquise. Pour le futur, la création de classe sera bien plus efficiente. Après tout, la programmation, peu importe le langage, est un entraînement constant et sur le long terme. Son apprentissage ne se finit donc jamais réellement.

En de rare moment, des décisions ont été prises de manière unilatérale menant à de courtes incompréhensions. Le manque d’expérience du travail d’équipe rend difficile de trouver l’équilibre en matière de communication : quand est-ce trop ? quand est-ce pas assez ?

## Marche à suivre

### Mise en place de la base de données dans HeidiSQL

#### Etape par étape

Avec le compte administrateur, créer la base de données centre\_medical\_DB, ainsi que ses tables avec le script **SQL\_script\_importDBv3.sql** qui figuredans le dossier **Database**.

Créer ensuite un utilisateur qui sera donc assigné à cette base de données en particulier.

* **Nom utilisateur** : admin\_centre\_med
* **Mot de passe** : Pa$$w0rd

Se connecter ensuite avec cet utilisateur et importer les données dans les tables.

Les données doivent être insérées à la suite avec les scripts figurant dans le dossier **Data** dans **Database**. Il faut importer les données dans l’ordre de numérotation des scripts 01 > 02 > 03 > … > 07.

#### Base de données complète

Il existe également un script appelé **centre\_medical\_DB.sql** qui comporte la création de la DB et des tables ainsi que les données des tables pour une importation plus rapide et en une fois.

L’utilisateur du point 1.6.1.1 doit également être créer de la même façon.

### Utilisation de l’application

Au lancement de l’application, une fenêtre de **Connexion** s’ouvre comme ci-dessous :



**Identifiant** : admin

**Mot** **de** **passe** : Pa$$w0rd

Une fois les données entrées correctement, un **ENTER** ou un clic sur **Se** **connecter** va ouvrir la fenêtre d’accueil de notre application de gestion d’un centre médical :

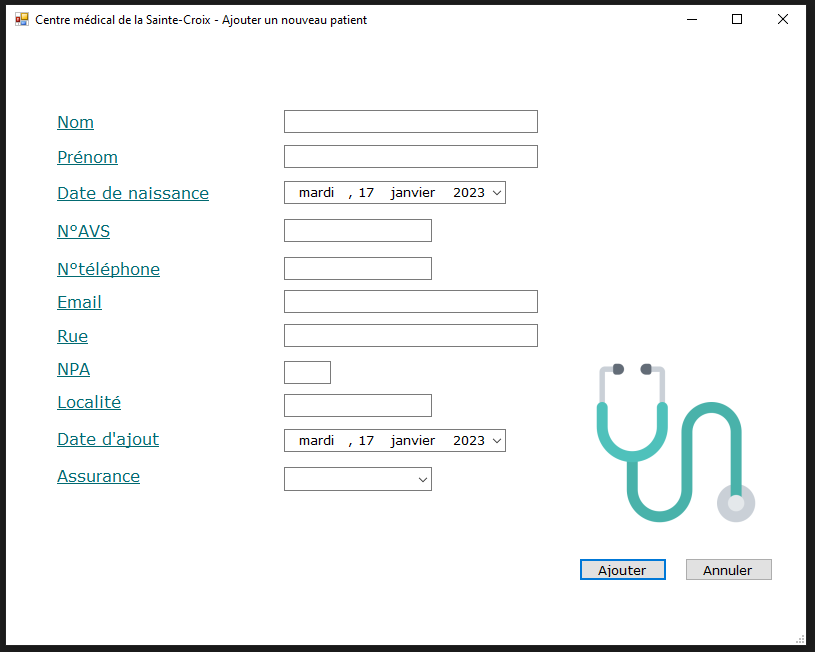
Depuis ici nous pouvons avoir accès à différentes fonctionnalités.

La 1ère option, **Ajouter un patient**, va ouvrir une nouvelle fenêtre (**Ajouter un nouveau patient**) se présentant comme un formulaire pour entrer les données personnelles d’un nouveau patient.

La 2ème option, **Rechercher un patient**, va ouvrir une nouvelle fenêtre (**Rechercher un patient**) nous montrant dans un DataGridView la liste des patients existants dans notre base de données.

La 3ème option, **Se déconnecter**, va tout simplement nous ramener à la fenêtre précédente de **Connexion**.

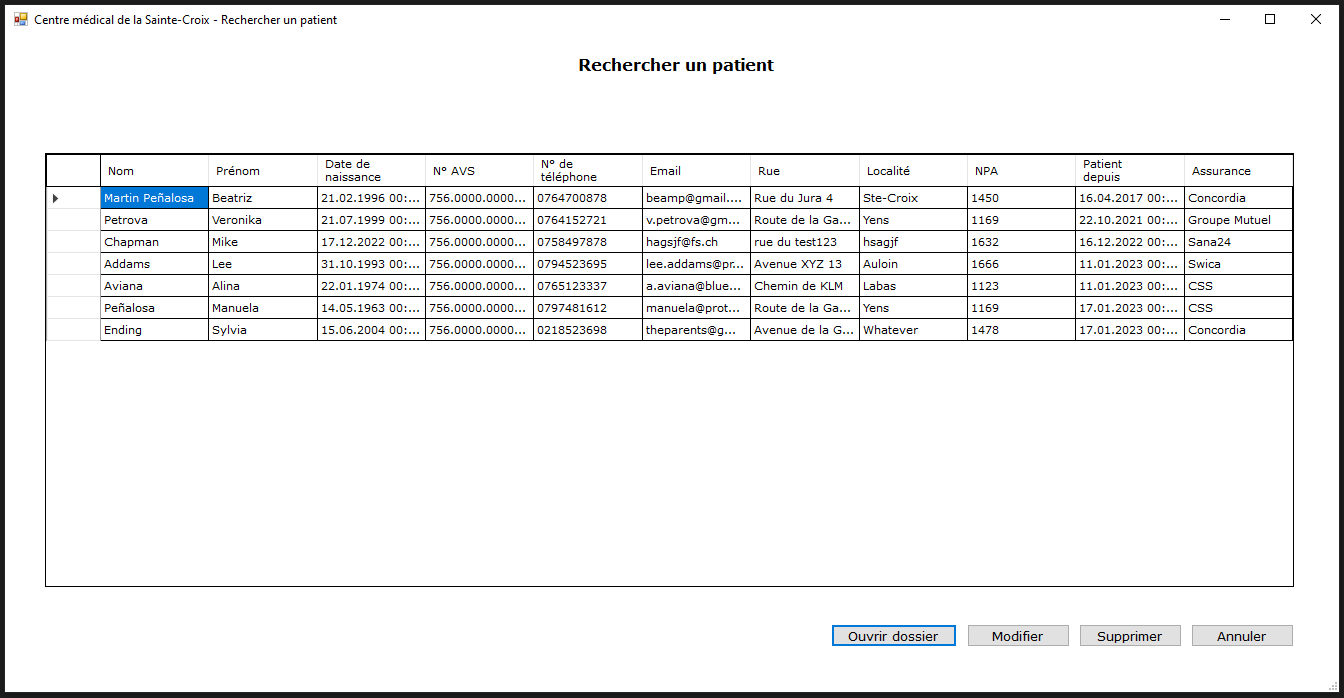
Attention, il faut appuyer sur l’icône correspondante et non sur le texte pour ouvrir la prochaine fenêtre.

Sur ce formulaire, **Ajouter un nouveau patient**, il faut donc entrer toutes les données indiquées avant de cliquer sur le bouton **Ajouter**. Les informations sont vérifiées, c’est-à-dire que si un champ est laissé vide, une fenêtre avertira l’utilisateur que des informations sont manquantes.

Si une information, comme par exemple le numéro AVS qui doit être unique, est dupliquée, à nouveau une fenêtre d’avertissement nous demandera de revérifier les informations.

Au clic du bouton **Ajouter**, les informations sont envoyées dans la base de données et le nouveau patient sera donc ajouté dans le centre médical.

Au contraire, si le bouton **Annuler** est cliqué, les données du formulaire ne sont pas envoyées, le formulaire se ferme et la **Page d’accueil** s’ouvre à nouveau.



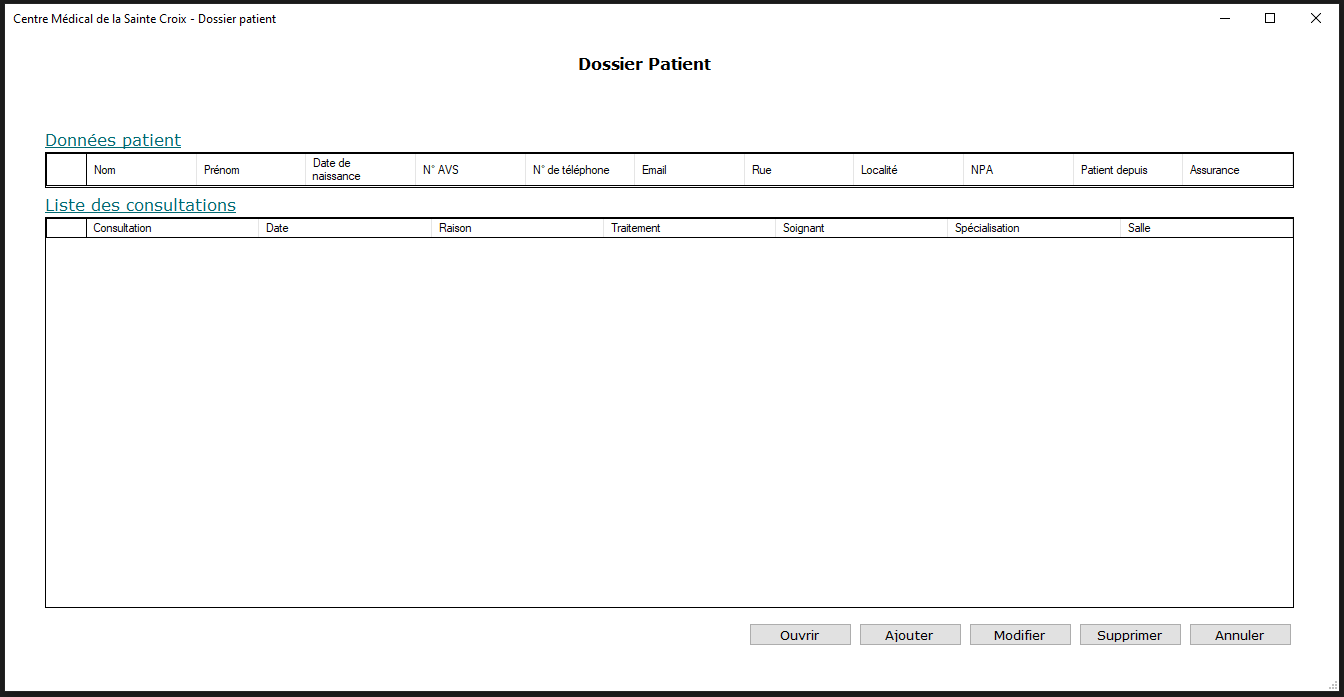
Ci-dessus, nous avons la fenêtre qui s’ouvre si l’icône de **Rechercher un patient** est cliqué dans la **Page d’accueil**. Ce DataGridView nous montre la liste de tous les patients ajoutés jusqu’à maintenant.

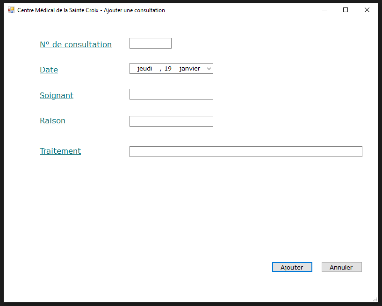


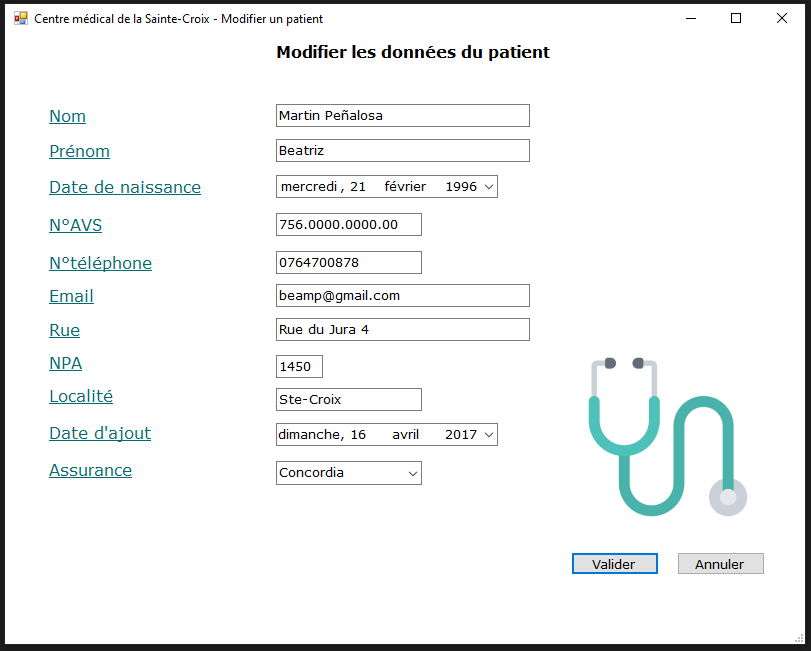
Pour effectuer une action sur un des patients de la liste, il faut appuyer sur la petite flèche au début de la ligne pour sélectionner en entier.

Le bouton **Supprimer** n’est pas implémenté.

Si le bouton **Annuler** est cliqué, la fenêtre se ferme et la **Page d’accueil** s’ouvre à nouveau.

Si le bouton **Ouvrir dossier** est cliqué, celui-ci ouvre une nouvelle fenêtre comportant les informations personnelles du patient, ainsi que la liste de ses consultations. Ceci n’a pu être implémenté au complet. 

En effet, la fenêtre **Dossier patient** va s’ouvrir, mais à partir de là, aucune information ne s’affiche et la plupart des boutons ne sont pas implémentés, sauf le bouton **Annuler** pour revenir au formulaire précédent **Rechercher patient** et le bouton **Ajouter**, qui ouvre le formulaire non fini d’ajout d’une nouvelle consultation **Ajouter une consultation**.

Revenons-en à la fenêtre de **Rechercher un patient**. Si le bouton **Modifier** est cliqué, celui-ci ouvre une nouvelle fenêtre comportant le même formulaire d’ajout de patient mais cette fois-ci avec les informations inscrites dans la base de données du patient sélectionné.

Ceci nous permet d’apporter des modifications en cas d’erreur ou de changement.

De même que pour l’ajout des patients, les données sont vérifiées (pas de champ vide, pas de donnée dupliquée).

Au clic du bouton **Valider**, les informations sont envoyées à la base de données pour modifier les anciennes données. Le formulaire se ferme et ouvre la fenêtre de **Rechercher un patient** avec la liste des patients ajoutés et/ou modifiés.

## Conclusion

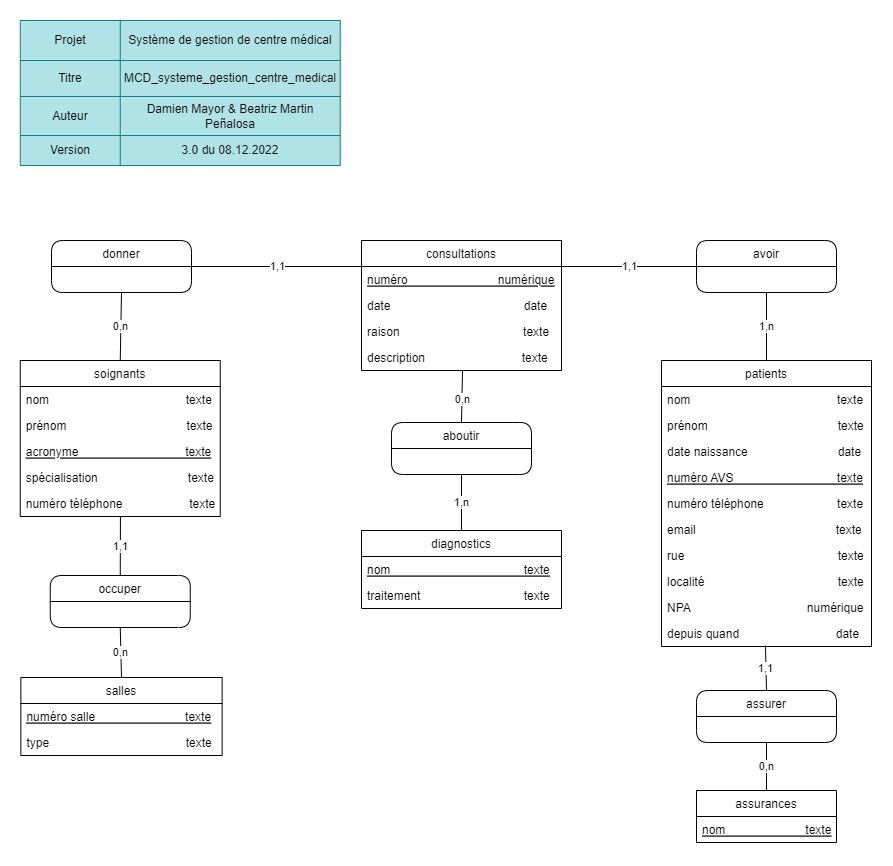
Ce projet est un des premiers gros projets que nous avons réalisé au sein du CPNV. La planification et le déroulement de celui-ci nous a été entièrement incombé. Il a donc fallu savoir faire preuve d’organisation et de communication pour mettre au point une bonne stratégie de travail. Bien entendu, ceci est toujours plus simple en théorie qu’en pratique. Notre manque d’expérience à divers niveaux s’est donc fait ressentir, mais tout de moins nous a permis d’apprendre de nos erreurs et de comprendre quels aspects d’un projet sont primordiaux.

De manière générale, ce projet a été une expérience très positive qui nous a fait découvrir de nouvelles notions et nous a fait approfondir d’autres notions vues durant notre apprentissage. De plus, il a associé divers cours donnés jusqu’à maintenant, ce qui a permis une mise en pratique intéressante et pas totalement inconnue.

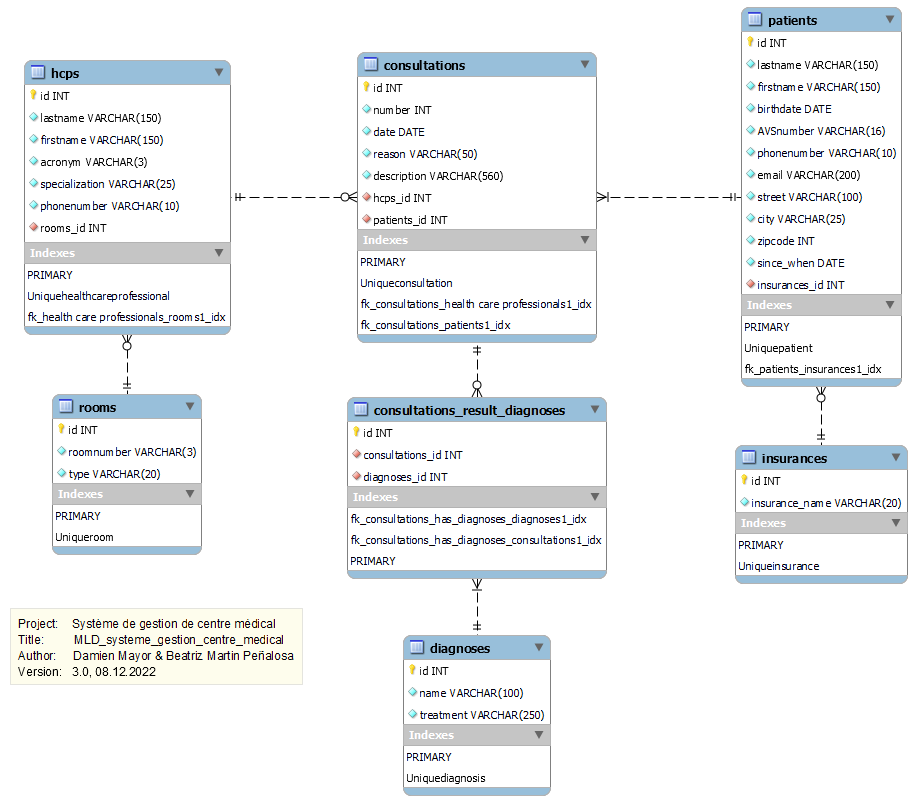
Nous aurions voulu pouvoir continuer plus longtemps pour aboutir notre projet, mais nous avons dû accepter que parfois ce n’est pas la destination qui compte, mais le voyage.

## **Annexe 1** : Modèles de données

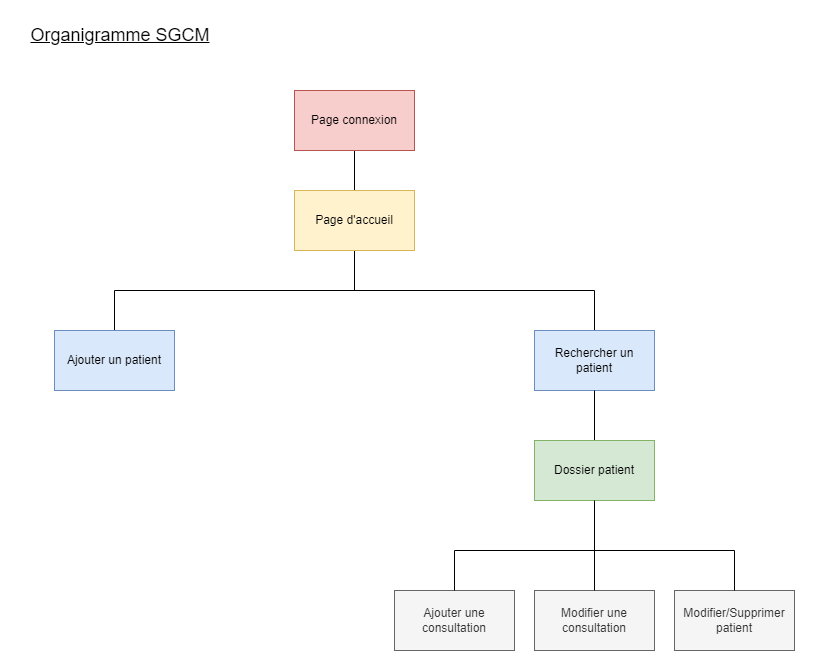
### MCD



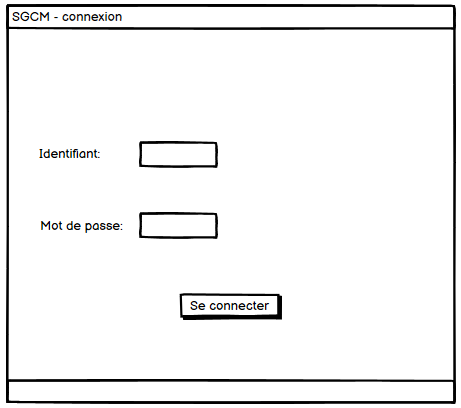
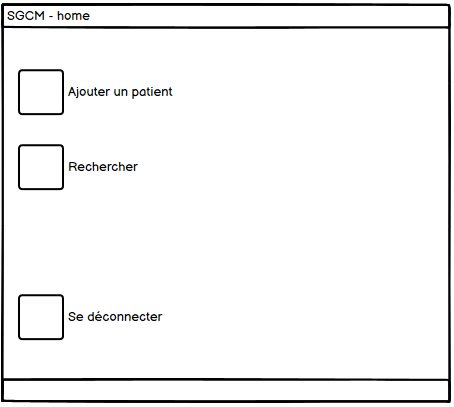
### MLD

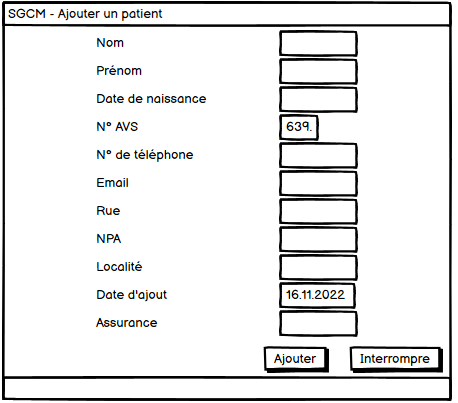
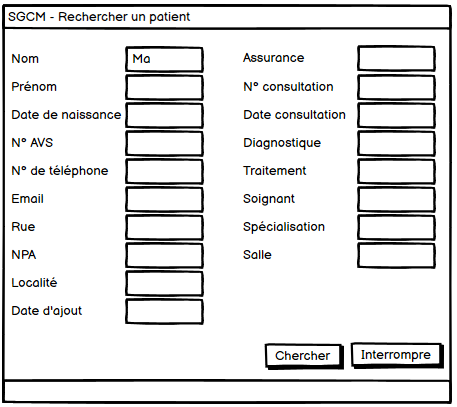


## **Annexe 2** : Organigramme

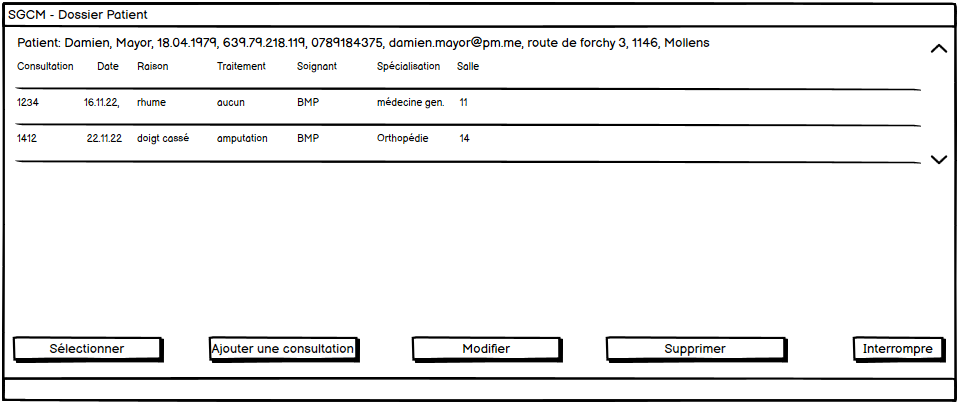


## **Annexe** **3** : Maquettes

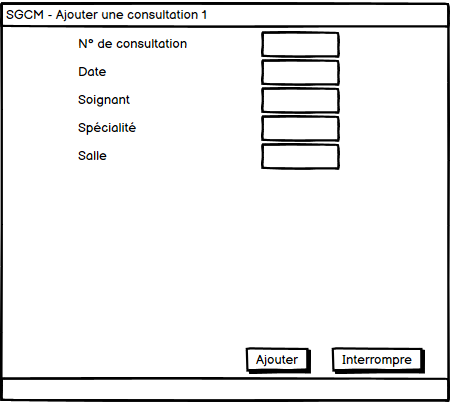
 





Une image contenant texte

Description générée automatiquement



Une image contenant texte

Description générée automatiquement

## **Annexe** **4** : Moodboard



### Apparence formulaires

* Police : Verdana
* Taille police :
  + Label : 12
  + TextBox : 10
* Couleur police :
  + Label : #026A71 (R :2 G :106 B :113)
  + TextBox : noir
* Taille fenêtres :
  + **Dossier patient** : 1050; 696
  + **Recherche patient :** 1550; 696
  + **Login et Homepage** : 523; 469
  + **Ajouter nouveau patient, ajouter une nouvelle consultation et modifier données personnelles du patient et modifier consultation** : 816; 649
* Couleurfenêtre:
  + BackColor : White

## **Annexe** **5** : Journal de travail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N°** | **Date** | **Tâches** | **Par** | **Commentaires** |
| 1 | 15.11.2022 | Création équipe, choix du sujet, MCD, MLD | Team |  |
| 1 | 15.11.2022 |  |  |  |
| 2 | 15.11.2022 | passage du MCD en version papier à version électronique | Beatriz | MCD sur drawio, explications dans un fichier texte |
| 2 | 15.11.2022 | Création journal de travail, création wireframe pour application | Damien | Excel pour journal de travail et balsamiq pour le wireframe |
| 3 | 16.11.2022 | Création BDD, création fichier Excel pour données, insertion données | Beatriz | insertion des données dans Excel et insertion des données de insurances et rooms dans DBB |
| 3 | 16.11.2022 | Continuation wireframe pour application | Damien |  |
| 4 | 22.11.2022 | correction du MCD | Beatriz | corrigé et validé par M. Ithurbide |
| 4 | 22.11.2022 | Aide au MCD et continuation wireframe | Damien |  |
| 5 | 22.11.2022 | correction du MLD, script SQL, création de queries | Beatriz | importation script MLD, ajout des données rooms, insurances et hcps dans la base de données, correction des données dans Excel |
| 5 | 22.11.2022 | Création diagramme pour application | Damien | Trop de temps perdu sur le wireframe sans diagramme. Décision prise de faire un diagramme afin de faciliter, plus tard, la fin du wireframe. |
| 6 | 23.11.2022 | création queries, entrée des données dans Excel | Beatriz | ajout des données patients et consultations dans DB, ajout de données de diagnoses dans Excel |
| 6 | 23.11.2022 | absent panne véhicule | Damien | - |
| 7 | 29.11.2022 | création queries, entrée des données dans Excel, création classes | Beatriz | ajout des données diagnostics dans la DB, exportation du script complet de la DB, début création du code C# (classe Patient, ConnectionDB et Hcps avec leurs attributs, accesseurs, constructeurs) |
| 7 | 29.11.2022 | finalisation diagramme + modification wireframe | Damien | Divers changements, corrections et mise au propre |
| 8 | 29.11.2022 | création classes | Beatriz | création des classes Consultation, Diagnostics, Insurance, Room avec leurs attributs, accesseurs et constructeurs |
| 8 | 29.11.2022 | finalisation diagramme + modification wireframe | Damien | Divers changements, corrections et mise au propre |
| 9 | 30.11.2022 | modification code ConnectionDB.cs, ajout formulaire de Login | Beatriz | création classe Login et LoginForm |
| 9 | 30.11.2022 |  | Damien |  |
| 10 | 06.12.2022 | absente | Beatriz | - |
| 10 | 06.12.2022 | Préparation documents pour remise à Mr. Ithurbide | Damien | Vérification documentation demandée et mise à jour |
| 11 | 06.12.2022 | absente | Beatriz | - |
| 11 | 06.12.2022 | Préparation documents pour remise à Mr. Ithurbide | Damien | Vérification documentation demandée et mise à jour |
| 12 | 07.12.2022 | mise au point, organigramme, réflexion moodboard | Beatriz | liste de questions pour développer application |
| 12 | 07.12.2022 | Mise au point, organigramme, réflexion moodboard | Damien | Vérification documentation demandée et mise à jour |
| 13 | 09.12.2022 | use case, mise à jour de la documentation, planification tâches, fichier JSON, mise à jour classe ConnectionDB, début formulaire page d'accueil | Beatriz | préparation documentation pour M.Ithurbide, fichier JSON pour se connecter à la DB |
| 13 | 09.12.2022 | Code Consultation form | Damien |  |
| 14 | 13.12.2022 | modification formulaire d'ajout du patient ainsi que le code associé | Beatriz | combobox dans formulaire pour ajouter les assurances listées dans notre DB |
| 14 | 13.12.2022 | Code Consultation form | Damien |  |
| 15 | 13.12.2022 | modification code ConnectionDB.cs et Insurance.cs + Patient.cs | Beatriz | récupération de l'id attribué aux assurances pour ajouter la clé étrangère à la table patients et ainsi pouvoir ajouter un nouveau patient à la DB |
| 15 | 13.12.2022 | Code ConsultationDetail form | Damien |  |
| 16 | 14.12.2022 | Récupération des patients sous forme de liste depuis DB, création formulaire recherche de patient | Beatriz | query SQL SELECT puis création liste pour ajouter chaque objet Patient, query SQL UPDATE pour mettre à jour les données d'un patient, ajout datagrid pour la recherche de patient |
| 16 | 14.12.2022 | Code ModifyConsultation form | Damien |  |
| 17 | 20.12.2022 | création d'une table users et connexion à l'application avec identifiant et mot de passe | Beatriz | table users dans DB, méthode Login() dans ConnectionDB, ajout attributs, constructeur, assesseurs à la classe Login, contrôle champs dans LoginForm |
| 17 | 20.12.2022 | Code ModifyConsultation form | Damien |  |
| 18 | 20.12.2022 | création formulaire modification des données d'un patient, centrer tous les formulaires lors de leur chargement, touche enter programmée pour valider l'identifiant et le mot de passe | Beatriz | tests pour sélectionner un patient de la datagrid, création formulaire devant s'ouvrir lors de la sélection d'un patient (non fonctionnel pour l'instant) |
| 18 | 20.12.2022 | Code PatientFile form | Damien |  |
| 19 | 21.12.2022 | tests connexion à l'app, mise au point avec Damien | Beatriz | mise en commun, discussion sur l'apparence des formulaires |
| 19 | 21.12.2022 | tests connexion à l'app, mise au point avec Beatriz | Damien | Tentative de mise en commun du code via Github un échec, création d'une solution de secours. |
| 20 | 10.01.2023 | modification apparence formulaires | Beatriz | uniformiser la police, taille, couleurs, etc. |
| 20 | 10.01.2023 | Recréation des formulaires créés précédemment | Damien | Suite à l'échec de la réunification des codes, mes formulaires et mon code est recréé manuellement |
| 21 | 10.01.2023 | travail sur le data grid pour pouvoir modifier les données personnelles d'un patient | Beatriz | correction de la requête SQL, récupération du bon patient dans la base de données pour que les modifications soient bien appliquées |
| 21 | 10.01.2023 | Recréation des formulaires créés précédemment. | Damien | Suite à l'impossibilité de merge les branches le 21.12.2022 |
| 22 | 11.01.2023 | finalisation modification du patient, tests ajout patient, gérer les erreurs lors de l'ajout | Beatriz |  |
| 22 | 11.01.2023 | Code PatientFile form | Damien | Nouveaux problèmes d'interaction entre Visual Studio et Github, impossible de travailler sur le même fichier sans avoir des problèmes/pertes. |
| 23 | 17.01.2023 | tests ajout et modification des patients pour mieux gérer les exceptions et diverses erreurs | Beatriz | correction du formulaire de modification car lorsque erreur affichée, fermait le formulaire, maintenant ok |
| 23 | 17.01.2023 | Divers commentaires code + nettoyage | Damien |  |
| 24 | 17.01.2023 | documentation, correction bug login pas vu jusqu'à maintenant | Beatriz | finalisation rétrospective, conclusion et marche à suivre de l'application, problème de data reader ne se fermant pas dans le login lançant une exception |
| 24 | 17.01.2023 | Documentation + préparation présentation | Damien | Création de la présentation du 23.1.2022 |
| 25 | 18.01.2023 | Documentation + préparation présentation | Beatriz | finalisation du Powerpoint pour la présentation orale |
| 25 | 18.01.2023 | Documentation + préparation présentation | Damien | Finalisation documentation |

## **Annexe 6** : Journal de bord

|  |  |  |
| --- | --- | --- |
| **Date** | **Evénement** | **Par** |
| 15.11.2022 | Début du projet développement: Système de gestion d'un centre médical | Beatriz, Damien |
| 22.11.2022 | MCD/MLD validés par M. Ithurbide | Beatriz, Damien |
| 08.12.2022 | Modification du MLD validé par M. Ithurbide | Beatriz, Damien |
| 14.12.2022 | Changement dans la recherche de patient, pas de formulaire pour entrer critères de recherche, mais un datagrid qui va afficher tous les patients entrés dans la base de données, car première idée trop complexe par rapport aux compétences acquises jusqu'à maintenant | Beatriz |
| 21.12.2022 | Mise en commun du code via Github un échec, solution de secours mise en place. | Beatriz, Damien |
| 10.01.2023 | Problèmes entre Visual Code et Github, impossibilité de travailler sur le même fichier sans créer des conflits. | Beatriz, Damien |
| 20.01.2023 | Remise dossier à M. Ithurbide | Beatriz, Damien |