

**Documentation technique**

Aide aux diabétiques

Lucas Pighini

cENTRE DE FORMATION PROFESSIONNELLE TECHNIQUE

|  |  |  |
| --- | --- | --- |
| **Date** | **Description** | **Version** |
| 22.05 | Page de garde | 1.0.0 |
| 22.05 | Versionning | 1.0.1 |
| 22.05 | Introduction | 1.1.1 |
| 22.05 | Fonctionnalités | 1.2.1 |
| 22.05 | Recommandations | 1.2.2 |
|  |  |  |

# Versionning

Table des matières

[Versionning 1](#_Toc515865365)

[Introduction 3](#_Toc515865366)

[Prérequis 3](#_Toc515865367)

[Fonctionnalités 3](#_Toc515865368)

[Connexion  3](#_Toc515865369)

[Accueil 3](#_Toc515865370)

[Mesures glycémie 3](#_Toc515865371)

[Graphiques de mesure 4](#_Toc515865372)

[Tableau de mesures 4](#_Toc515865373)

[Tableau d’insuline 4](#_Toc515865374)

[Profil utilisateur 4](#_Toc515865375)

# Introduction

Etant actuellement en 3ème et dernière année accélérée de ma formation en informatique au sein du Centre de Formation Professionnel Technique, j’ai pour but de réaliser un travail pratique individuelle afin de confirmer l’obtention de mon CFC. J’ai donc choisi de développer une application C# qui permettra aux patients diabétiques et à leur médecin d’avoir un suivi informatisé et simplifié de leurs mesures de glycémie.

# Motivation

Au milieu de ma 3ème année, j’ai dû décidé de ce que je voulais faire comme projet pour mon TPI. D’abord j’ai dû choisir entre une application Web ou C#, ayant découvert Bunifu qui est une librairie de composant graphique pour Visual Studio, j’ai donc pris le choix de développer en C#. Ensuite il a fallu que je réalise un cahier des charges de ce que je voulais faire comme application, n’ayant pas vraiment d’idée je me suis donc décidé à faire une application qui pourrait se rendre utile pour un membre de ma famille. Ma grand-mère étant diabétique, j’ai donc pu voir plusieurs fois ma mère s’occuper des mesures de la glycémie et j’ai pu remarquer que tout est noté à l’écrit dans un carnet. C’est pourquoi j’ai décidé de simplifier les mesures et de les rendre informatisées.

# Cahier des charges

## Objectif

L’objectif de cette application Windows est de permettre à un patient d’entrer ses mesures quotidiennes de glycémie et ensuite de pouvoir visualiser les données.

## Fonctionnalités

Les utilisateurs connecté en tant que médecin auront les fonctionnalités suivantes :

* Consulter sous forme tabulaire les mesures d’un patient
* Sur le graphique des mesures, il peut choisir la date de début et de fin et les types
* Afficher sur un graphique la tendance des mesures de glycémie
* Affichage tabulaire des mesures
* Mesure de glycémie en lecture seul
* Gestion complète du tableau de recommandation d’insuline

Pour ceux qui sont connectés en tant que patient :

* Gestion complète des mesures de glycémie
* Sur le graphique des mesures, il peut choisir la date de début et de fin et les types
* Afficher sur un graphique la tendance des mesures de glycémie
* Affichage tabulaire des mesures et de la tendance
* Gestion complète du tableau de recommandation d’insuline

# Analyse de l’existant

# Analyse fonctionnelle

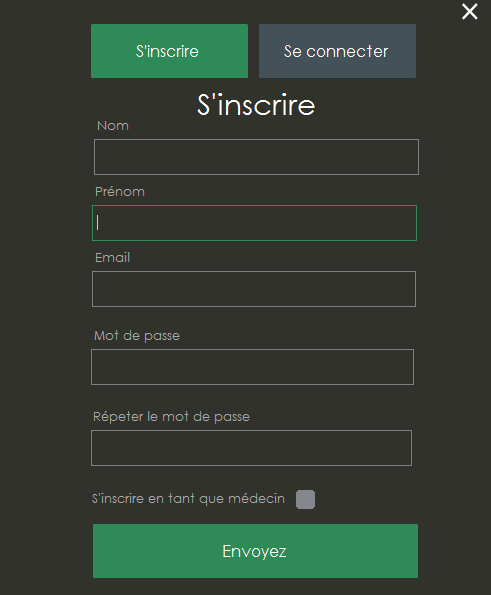
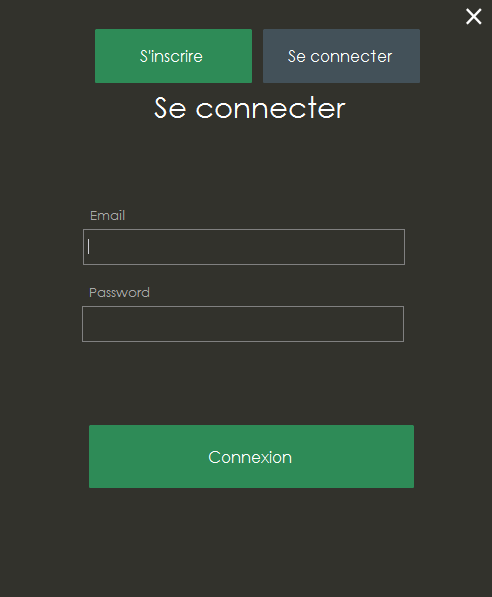
Ce programme va contenir plusieurs fonctionnalités qui offriront à l’utilisateur une aide et un suivi sur l’évolution de son diabète. Les fonctionnalités seront différentes entre un utilisateur identifié en tant que patient qu’un médecin.

## Connexion

Cela sera la première page à s’afficher lors du démarrage de l’application

L’utilisateur aura la possibilité de se créer un compte

Après avoir créé son compte l’utilisateur pourra se rendre sur « Se connecter » et se connecter à l’application

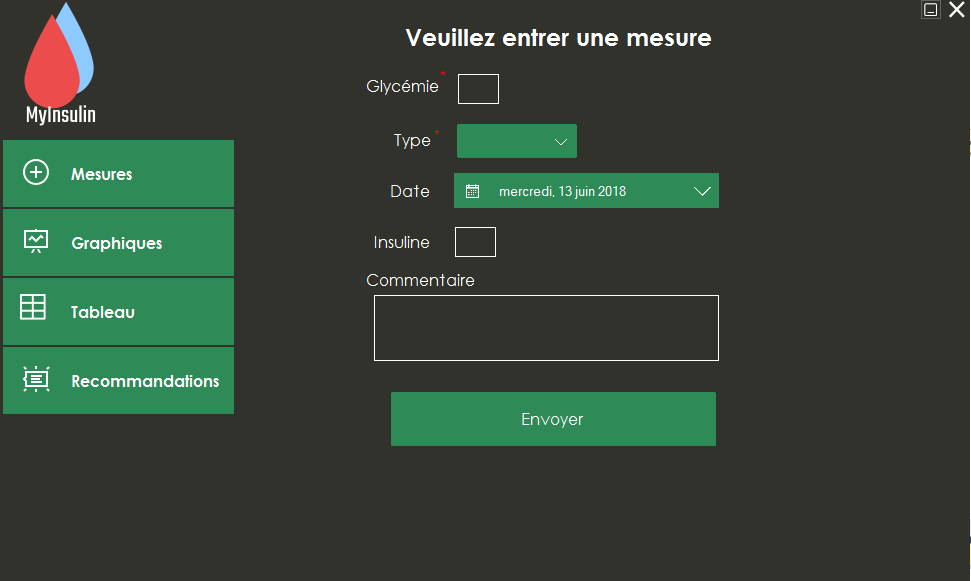


## Mesures glycémie

L’utilisateur aura la possibilité sur cette page de créer de nouvelles mesures de glycémie.

S’il valide et envois la mesure à la base de donnée, il recevra une recommandation pour savoir quel niveau d’insuline il doit s’injecter.

Dans la page « Tableau de Mesure », quand l’utilisateur cliquera sur une donnée il sera redirigée sur cette page pour modifier la mesure déjà existante.

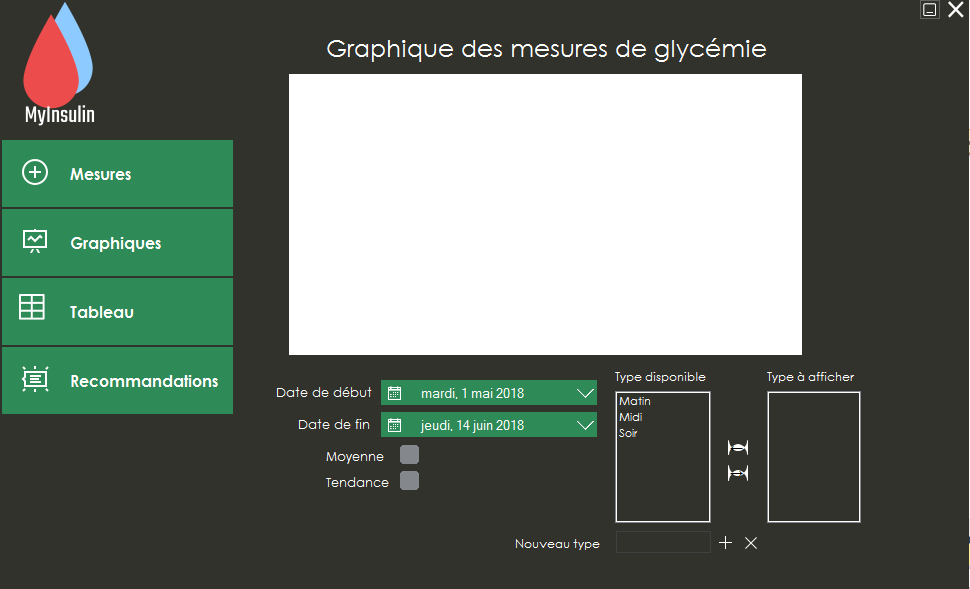


## Graphiques de mesure

Sur cette fenêtre, il sera possible de consulter ses mesures à partir d’un graphique. L’utilisateur peut choisir la date de début et de fin.

Il faut aussi laisser à l’utilisateur le choix de trier les différents types de courbe à afficher

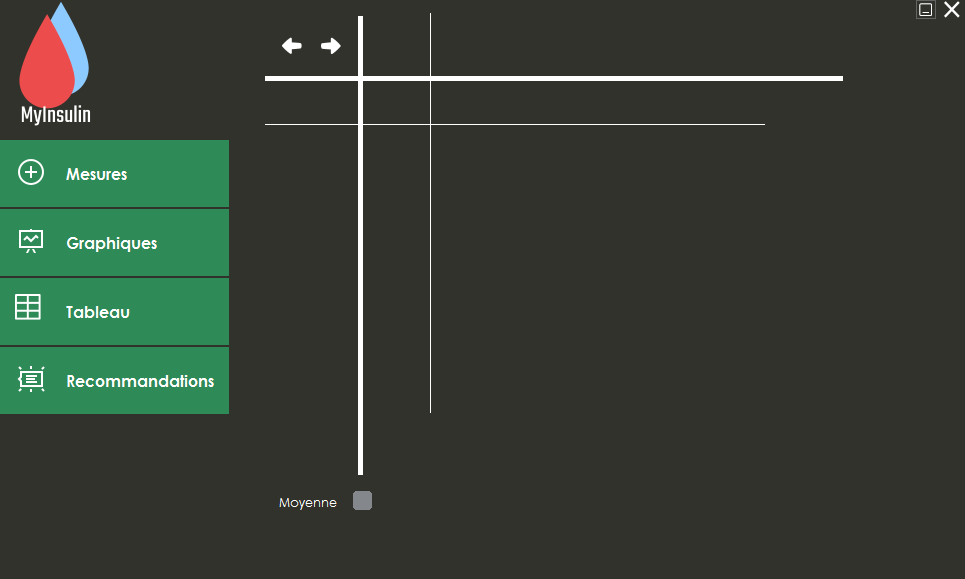
Et pour finir il peut aussi afficher ses données sous la forme d’une courbe de tendance



## Tableau de mesures

L’utilisateur aura aussi la possibilité de consulter ses mesures sous forme tabulaire.

Les données qui seront affiché seront toutes celles qui remonte à 7 jours avant la dernière mesure faites. Il aura la possibilité de se promener de semaine en semaine.



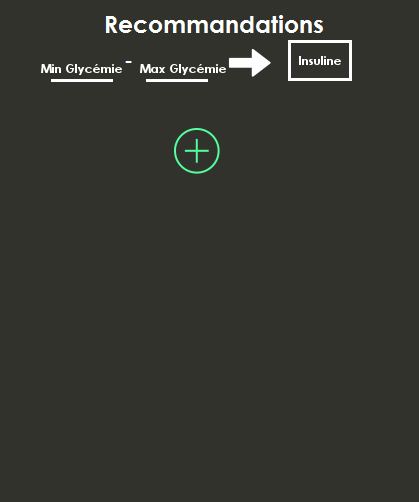
Si une valeur doit être changé, il suffit de cliquer sur la mesure et l’utilisateur sera redirigé sur la page de « Gestion de mesure ».

## Tableau d’insuline

Cette page permettra de changer le tableau de recommandation d’insuline d’un patient.

Il pourra ajouter, modifier et supprimer de nouvelles recommandations.

Ce tableau servira à recommander correctement l’utilisateur lorsqu’il entrera une nouvelle donnée.



## Ajout d’un patient

L’utilisateur aura un accès à cette page uniquement s’il s’est connecté en tant que médecin.

Cette page lui permettre d’ajouter à un patient et de pouvoir accéder à son suivi.



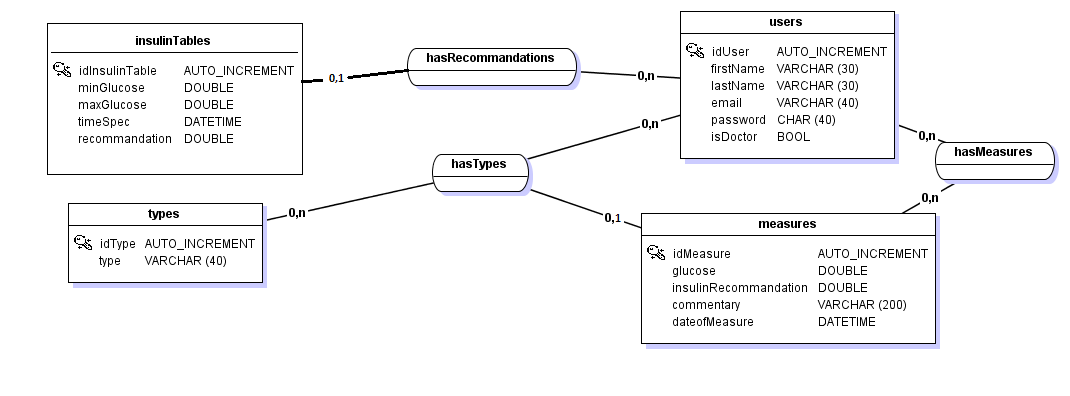
# Analyse organique

## Framework

Pour réaliser l’interface de cette application je me suis servie d’une librairie de composant graphique C# nommée Bunifu. Cela me permet de créer un interface plus ergonomique et plus design que ce qu’il y a de base sur Visual Studio. Pour utiliser cette bibliothèque il faudra payer une somme de 250$ pour une année d’utilisation sur Visual Studio, ayant découvert cette bibliothèque il y a de ça 7 mois je l’ai acheté quand il ne coutait encore que 50$.

## Base de données

Pour réaliser cette base de données je me suis inspiré du MCD fournies dans le cahier des charges et j’y ai ajouté une relation entre la tables `users`, `measures` et `types`



En développant le MCD nous aurons les 5 tables suivantes :

### Users

La table « users » contient les informations personnelles de l’utilisateur : son nom (`lastName`), son prénom (`firstName`) et son `email`, ces trois champs sont en Varchar. Elle contient aussi le mot de passe qui est de type de char et d’une longueur de 40 car le mot de passe est haché en sh1 pour éviter de laisser paraitre en clair dans la base de données. Ensuite vient `isDoctor` qui est de type boolean et permet de déterminer si l’utilisateur est un médecin ou un patient. Et pour finir il y aura le champ `idMedecin` qui sera remplie par l’id du médecin d’un utilisateur. Cela restreint un médecin par utilisateur.

### Measures :

Cette table est composée des différentes valeurs nécessaire qu’un patient diabétique se doit de remplir lorsque qu’il prend ses mesures. Il y a comme champ `glucose`, qui représente le niveau de glucose dans le sang d’une personne et `recommandationInsulin` qui donne la quantité d’insuline par rapport à un taux de glycémie. Ensuite vient le champ `dateofMeasure` pour fixer une date à la mesure. Et pour finir un champ `commentaire ` si l’utilisateur voudrait laisser une information complémentaire.

### InsulinTable :

La table représente les recommandations d’insuline d’un patient. Elle se compose des champs suivants. Il y’a d’abord `minGlucose` qui est le minimum de glycémie, puis `maxGluxose` qui lui est le max de glycémie et qui à eux deux forment un interval. Et ensuite il y a `recommandation` qui donne la recommandation d’insuline par rapport aux deux valeurs définies précédemment. Ainsi que `timeSpec` qui donne la date à la quel elle a été créé ou modifié et pour l’utilisateur qui a fait cette recommandation.

### Types :

Cette table permet de stocker les différents types de période horaire qu’un utilisateur souhaite ajouter à ses mesures Elle se compose d’un champ `type` qui représente le nom du de la période. Cela permet d’avoir plusieurs types différents et non pas des valeurs fixées

### HasType

Cette table est créé par la relation « 0,n/0,n » en la table `users` et `types`, elle est composée des id des deux tables. Cette table permet de définir des types à un utilisateur et qu’il puisse gérer ses propres types.

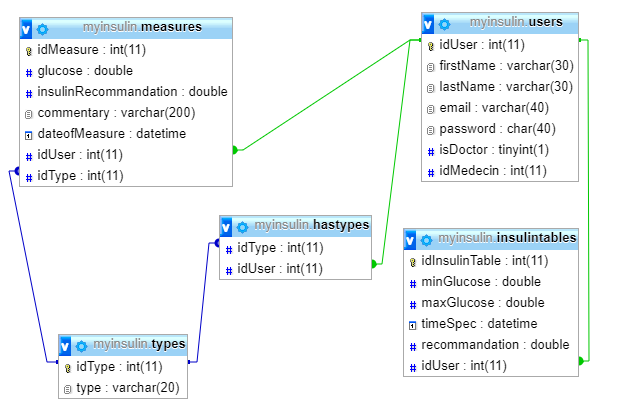
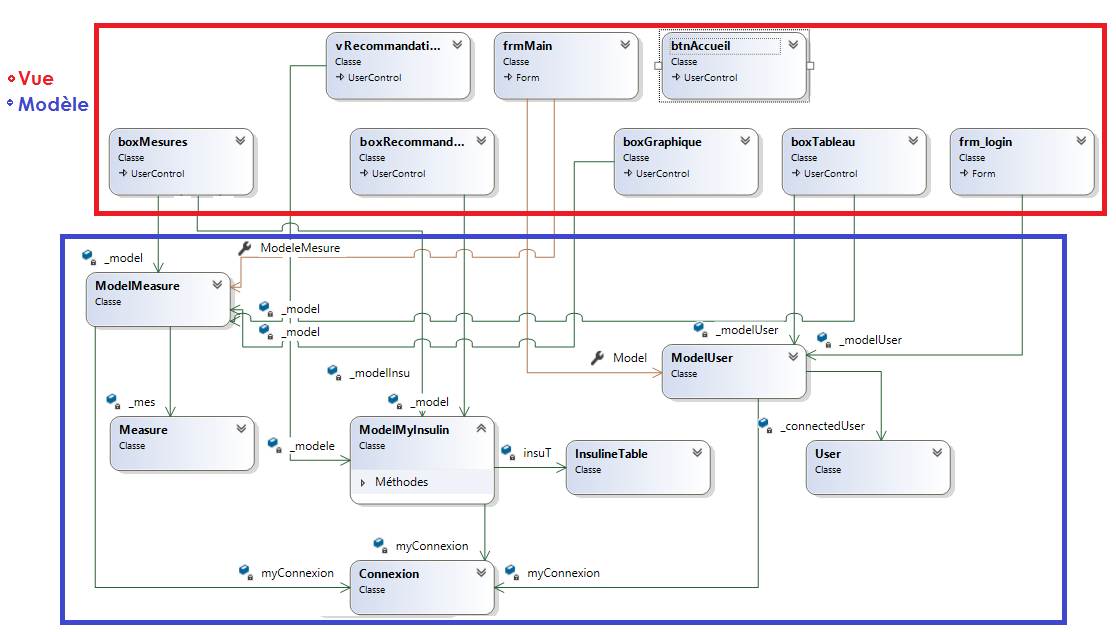


Figure Vue depuis le concepteur de PhpMyAdmin

## Modélisation

Pour le développement j’ai opté pour une architecture Modèle-Vue, le modèle ne s’occupant uniquement des données et la vue doit uniquement afficher correctement les données fournies par le modèle. Voici un diagramme de classe avec une représentation visuelle du MV que j’ai mis en place pour ce projet :



## frmLogin

La classe « frmLogin » hérite du type « Form »et apparatient donc à la vue car elle va s’occuper d’afficher la fenêtre d’inscription ou de connexion à l’utilisateur. Elle va instancier le « ModelUser » pour lui envoyer les données des formulaires, lorsque l’utilisateur se connecte il va passer son « ModelUser » en paramètre au « frmMain ».

## frmMain

Cette classe hérite aussi du type « Form », elle est la fenêtre principale et va s’occuper d’afficher les différentes vues. Les fenêtres qu’elle doit afficher sont des contrôles utilisateurs de Visual Studio que j’ai créé moi-même qu’elle va ajouter ou enlever de son panel principal. Lorsque « frmMain » est instancié, il reçoit en paramètre le « ModelUser » et instancie un nouveau « ModelMeasure ». La page se doit de gérer un affichage différent lorsque l’utilisateur est un médecin ou bien un patient d’où l’utilité d’un modèle gérant les utilisateurs.

## boxMeasure

Ceci est un contrôle utilisateurs que j’ai créé afin de simplifier l’affichage sur la page principale. Cette classe permet à un utilisateur de saisir les différentes données qui compose une mesure de glycémie d’un diabétique.