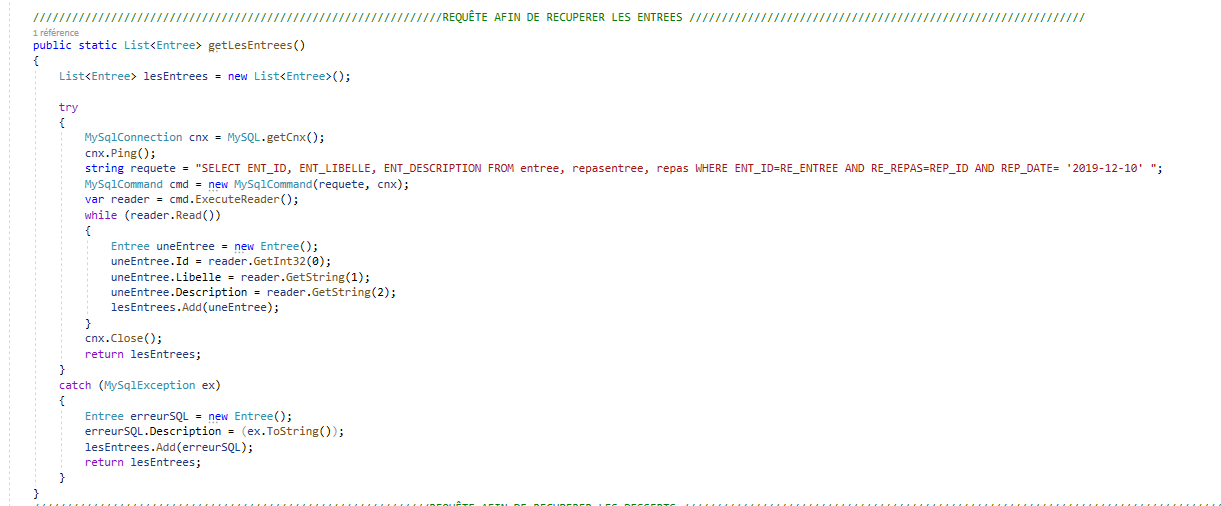
|  |  |
| --- | --- |
|  |  |
|  |  |

MyCantine récupération et affichage des données

Dans ce fichier on peut retrouver toutes les requête relative à l’affichage des données telle que le listing des menus proposés pour une journée ou l’affichage de récapitulatif des choix de l’utilisateur.

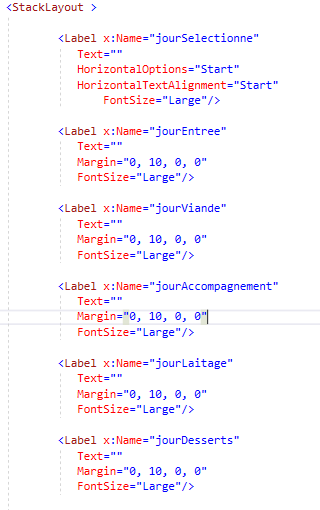
Exemple d’affichage des entrées proposé pour un jours donné.

Dans le fichier MyCantineSQL.cs

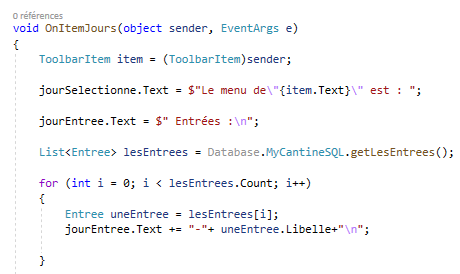


Ici on fait une requête select SQL pour récupérer les entrées en fonction d’une date. On hydrate la liste nommée lesEntrees avec le résultat de la requête. Une fois que le résultat de la requête est stocké dans la liste on peut l’utiliser comme on le souhaite.

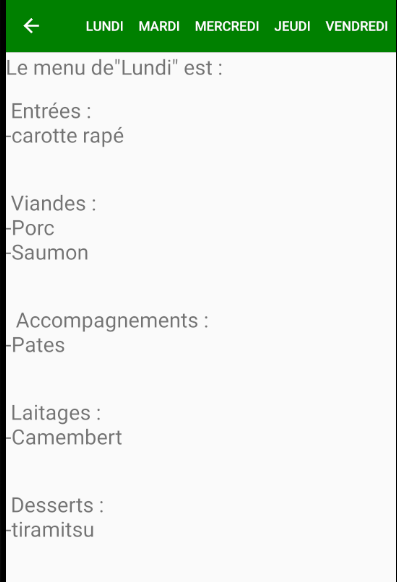
On déclare ici les zones où on veut afficher les résultats de la requête.



Afin de récupérer le résultat de la requête pour l’afficher on utilise une boucle qui va parcourir la liste et en afficher tout les resultat dans les zones correspondantes.



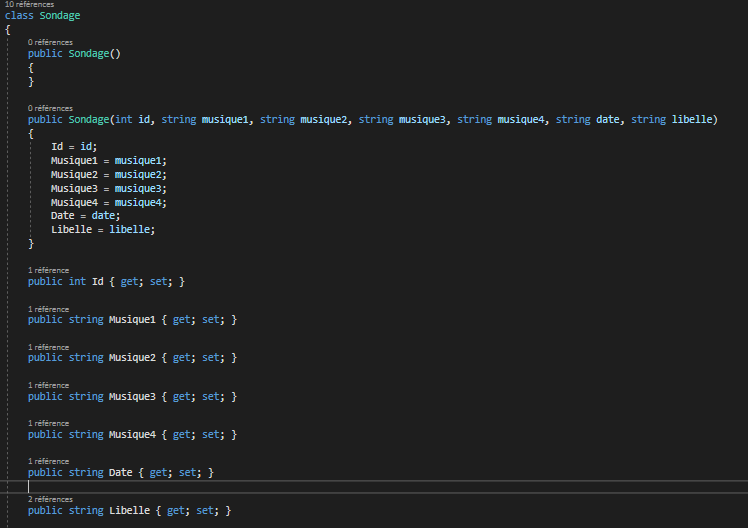
Le résultat :



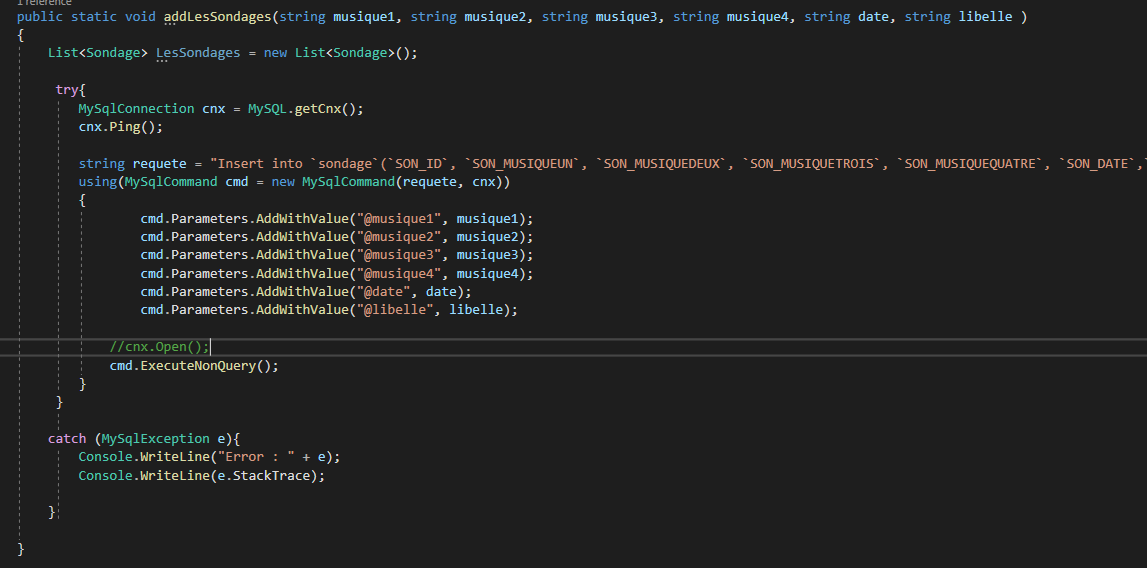
MySonnerie :

AJOUT D’UN SONDAGE D’UNE SONNERIE :

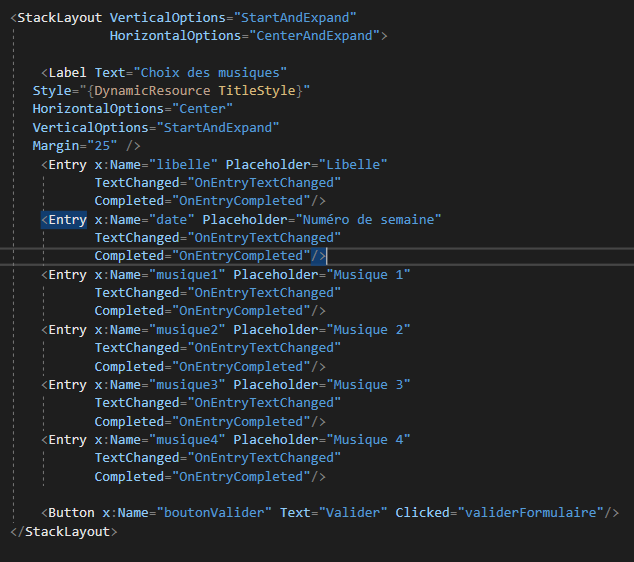
Models.Sondage.cs -> constructeurs du sondage.



Dans database.MySonnerie.cs fonction d’ajout d’un sondage dans la base de données (pensez à changer les champs)



FormulaireAjoutSonnerie.xaml -> vue d’ajout d’un sondage d’une sonnerie.



C# du formulaire d’ajout d’un sondage. Fonction pour envoyer les données entrées à la fonction d’insertion des données de database.MySonnerie.cs



Fonction de vérification afin que tous les champs soient remplie avant d’envoyer dans la base de données.



Ce qu’il reste à faire pour l’ajout d’un sondage :

Après le pop up de l’ajout du sondage réussi, ramener à la page d’accueil.

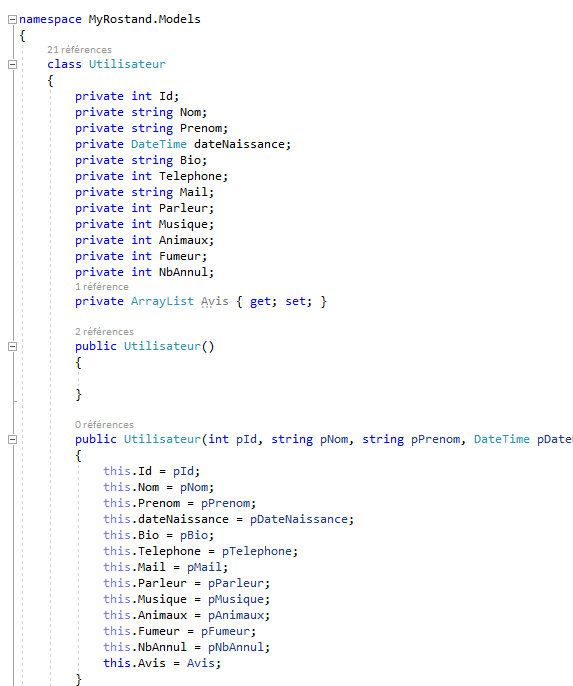
Dans l’application PHP :

* lister les sondages proposé par les éleves (libelle, date)
* ajouter un sondage (admin ou membre du CVL)
* consulter un sondage
* modifier un sondage
* supprimer un sondage

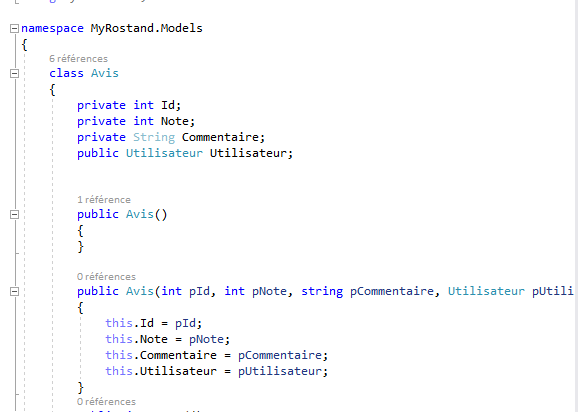
MyCovoit:

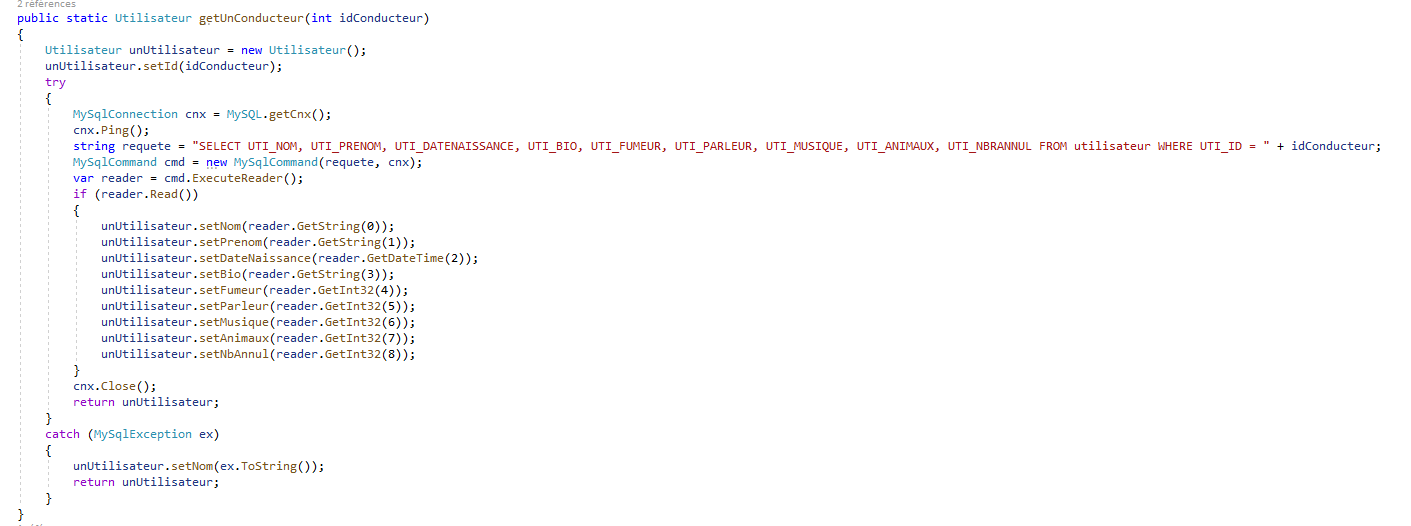
Gestion du Profil Utilisateur dans la section MyCovoit de MyRostand METZ Julien

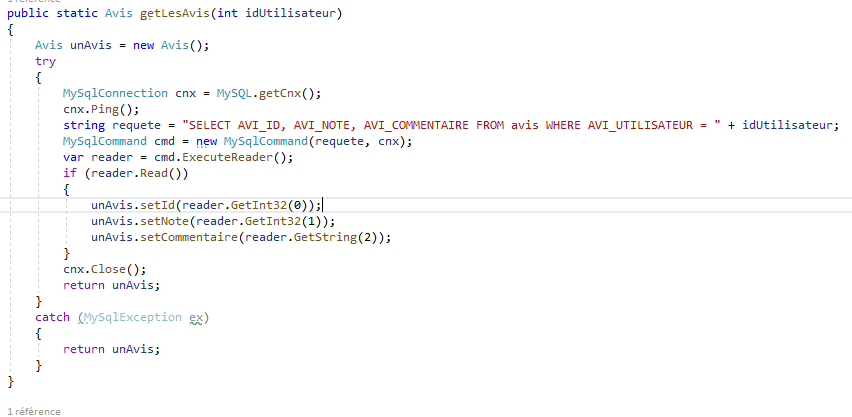
Utilisateur.cs permet de représenter toutes les données de la table utilisateur.



Avis.cs permet de représenter toutes les données de la table avis.



Pour récupérer les données dans la base de données on créer dans le fichier MyCovoit qui se trouve dans le dossier DataBase. On créer une méthode pour récupérer un utilisateur(getUnConducteur) avec une requête qui sélectionne seulement et une méthodequi récupère les avis(getLesAvis).



Pour afficher les données qui sont rattachés à un utilisateur, cela ce passe sur le fichier ProfilUser.xaml dans le dossier MyCovoit.



Dans chaque label on a x:Name="" qui est le nom du champs et le Text="" qui contient ce qui sera affiché sur la page.

Puis dans le ProfilUser.xaml.cs, on appelle la méthode puis on récupère le nom des champs et on leur affecte la donnée à afficher.



Pour la variable Fumeur, Parleur, Musique ou Animaux qui sont des données soit 1 soit 0 on fait une fonction qui affiche une message selon la valeur de la variable.Pareil pour la note dans les avis.

