Projet SQL



Sean Ford – Cin2b

Lausanne ETML

9 Semaines

Monsieur Lopes

Table des matières

[1 Spécifications 3](#_Toc308526316)

[1.1 Titre 3](#_Toc308526317)

[1.2 Description 3](#_Toc308526318)

[1.3 Matériel et logiciels à disposition 3](#_Toc308526319)

[1.4 Prérequis 3](#_Toc308526320)

[1.5 Cahier des charges 3](#_Toc308526321)

[1.5.1 Objectifs et portée du projet (objectifs SMART) 3](#_Toc308526322)

[1.5.2 Caractéristiques des utilisateurs et impacts 3](#_Toc308526323)

[1.5.3 Fonctionnalités requises (du point de vue de l’utilisateur) 3](#_Toc308526324)

[1.5.4 Contraintes 3](#_Toc308526325)

[1.5.5 Travail à réaliser par l'apprenti 4](#_Toc308526326)

[1.5.6 Si le temps le permet … 4](#_Toc308526327)

[1.5.7 Méthodes de validation des solutions 4](#_Toc308526328)

[1.6 Les points suivants seront évalués 4](#_Toc308526329)

[1.7 Validation et conditions de réussite 4](#_Toc308526330)

[2 Planification Initiale 4](#_Toc308526331)

[3 Analyse 4](#_Toc308526332)

[3.1 Opportunités 4](#_Toc308526333)

[3.2 Document d’analyse et conception 4](#_Toc308526334)

[3.3 Conception des tests 5](#_Toc308526335)

[3.4 Planification détaillée 5](#_Toc308526336)

[4 Réalisation 5](#_Toc308526337)

[4.1 Dossier de Réalisation 5](#_Toc308526338)

[4.2 Modifications 5](#_Toc308526339)

[5 Tests 5](#_Toc308526340)

[5.1 Dossier des tests 5](#_Toc308526341)

[6 Conclusion 5](#_Toc308526342)

[6.1 Bilan des fonctionnalités demandées 5](#_Toc308526343)

[6.2 Bilan de la planification 5](#_Toc308526344)

[6.3 Bilan personnel 5](#_Toc308526345)

[7 Divers 6](#_Toc308526346)

[7.1 Journal de travail 6](#_Toc308526347)

[7.2 Bibliographie 6](#_Toc308526348)

[7.3 Webographie 6](#_Toc308526349)

[8 Annexes 6](#_Toc308526350)

# Spécifications

## Titre

Se connecter à une base de donnée

## Description

Créer une application en Windows forms, qui permet de se connecter à une n’importe qu’elle base de donnée et d’y faire des actions comme Ajouter/Supprimer une base de donnée ou Ajouter /Supprimer une table.

## Matériel et logiciels à disposition

Ordinateur et logiciel microsoft

## Prérequis

Module 105

## Cahier des charges

### Objectifs et portée du projet (objectifs SMART)

### Travail à réaliser par l'apprenti

Créer une application fonctionnelle avec du C# et SQL. Pouvoir se connecter à n’importe qu’elle base de donnée et d’y faire des actions comme Ajouter/Supprimer une base de donnée ou Ajouter /Supprimer une table. Utiliser un Tree View pour afficher les différentes bases de donnée

## Les points suivants seront évalués

* Le rapport
* Le journal de travail
* Le code et les commentaires

## Validation et conditions de réussite

* Compréhension du travail
* Possibilité de transmettre le travail à une personne extérieure pour le terminer, le corriger ou le compléter
* Etat de fonctionnement du produit livré

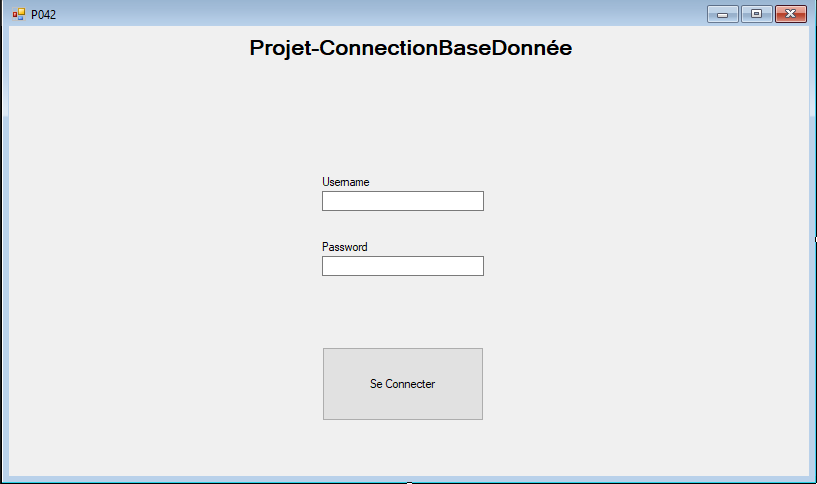
# Planification Initiale

Ce paragraphe présente le planning d'origine (date de début, date de fin, vacances et congés, liste hiérarchique des tâches ou GANTT, jalons, durée totale)

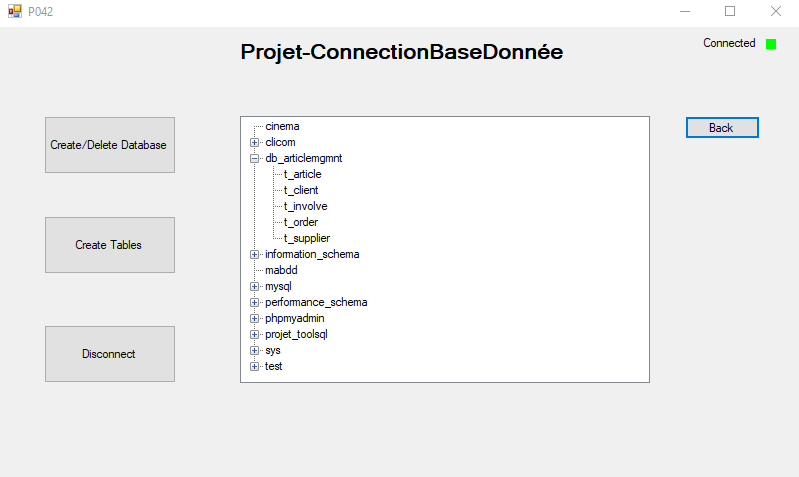
Toutes les mises à jour subies par le planning sont à reporter (avec date de mise à jour) et peuvent déboucher sur plusieurs versions de plannings.

# Analyse

## Document d’analyse et conception



J’ai décidé de faire la page de connexion comme ça car, pour le point de vu d’un utilisateur, c’est plus facile de rentrer des données sur une interface graphique que directement changer dans le code.

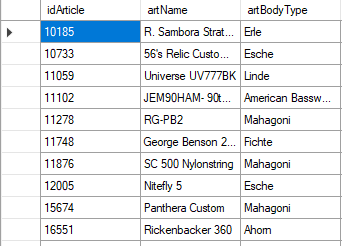


Voici ma page d’accueil, on peut créer/supprimer une table ou une Database.

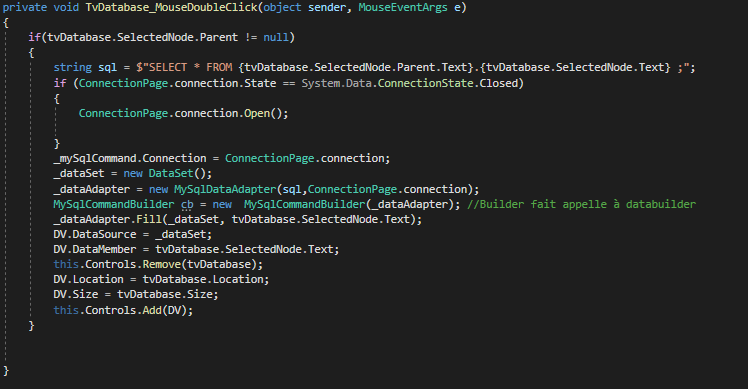
Au centre de l’écran il y a un Tree view ceci permet d’afficher de manière \*Tree\* toute les bases de données de l’utilisateur. J’ai choisis « Tree view » car c’est la manière le plus propre et c’est très simple à comprendre. Voir code si-dessous.



J’ai créé une instance avec le « TreeNode tn ; ». Avec « \_mySqlCommand.commandText = « show Databases ; » ; ». Puis je vérifie si l’utilisateur est toujours connecté à la base de donnée, et c’est seulement après cela que je commence le « Tree view ». Dans le code j’utilise un « ExecuteReader() » il permet de de faire la requête et de la stocker. Mais il ne peut qu’être utiliser une fois par requête SQL. Il faut faire un « \_mySqlDataReader.Close()> pour pouvoir réutiliser un « ExecuteReader() »



J’ai aussi fait un « Grid view » pour afficher les données dans une table. J’ai choisis cette technique car c’est la plus simple à faire et je trouve que sa donne bien.



Tout d’abord j’ai créé un évènement « double click », quand l’utilisateur double cliquera sur quelques choses dans le Tree view cela déclenchera la fonction.

Ensuite je regarde si le parent n’est pas égal = « NULL ». Après je fais le requête SQL. J’ai créé une instance avec le « \_dataSet = new DataSet () ». Je donne ma requête au dataAdaptater (Il faut l’instancier). Je lui donne la requête et la connexion

Ensuite le builder fait appelle au dataAdapteter. Pour finir j’utilise « FILL » et je donne au adapter le « \_dataSet », Le parent utilisé (la database double cliquée).

## Conception des tests

Comme test dans le code j’essaie de vérifier un maximum la connexion à la database. Car sinon le code ne fonctionne pas.

## Planification détaillée

A ce stade, après l’analyse complète du projet, un planning détaillé et complet (avec tâches, sous-tâches, dépendances, durée, …) peut être finalisé.

Le planning détaillé doit s’inscrire dans le planning initial. Il faut que l’on puisse situer cette planification détaillée par rapport à la planification initiale.

# Réalisation

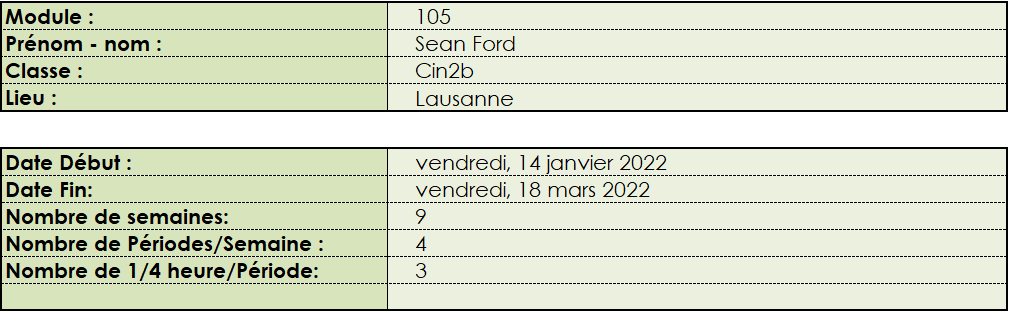
# Conclusion

## Bilan personnel

SI c’était à refaire je changerai l’ordre dans laquelle j’ai fait les pages autrement je trouve que j’ai beaucoup appris

# Divers

## Journal de travail



## 

Voici comment mes périodes ont été utilisée.