

# Compte rendu par semaine

## Semaine 1

Durant la première semaine de stage, j'ai pu faire une première réunion avec ma maître de stage, Céline Batais, la présidente du club de basket **GASB05**. Cette première réunion m'a permis de déterminer les problèmes au sein du site du club mais aussi d'en savoir plus sur la demande principale de Céline Batais, à savoir d'avoir un système de mail automatique qui envoie un mail à tous les licenciés du club dont l'anniversaire est le jour même.

### Les problèmes du site

Les problèmes du site qui ont été constatés sont principalement des problèmes de visibilité : Il est beaucoup moins visité que le facebook du club car il est plus compliqué de trouver les éléments recherchés dessus. Or ce site coûte 150€ à l'année et entraîne donc des pertes.

Le second problème est qu'il n'inclut pas dans son offre de service php et de base de données MySQL. Il faudrait payer 27€ de plus, or, je ne souhaite pas faire payer un supplément au club qui est déjà en perte. J'ai donc trouvé le site Lixia.li afin de trouver une alternative gratuite au pack proposé par Quomodo.

### L'envoi automatique de mail

J'ai construit ce système avec un script php qui se connecte à une base de données MySQL et qui y récupère les adresses mail et les prénoms des licenciés du club dont c'est l'anniversaire le jour même. Cependant, je n'ai ni base de données, celle du club étant une BDD Excel, pas beaucoup d'expérience en php, je ne connais, à ce moment, aucun système qui permet de faire tourner un script php une seule fois par jour sans avoir à l'appeler nous même.

Je décide donc de commencer par rechercher comment faire une BDD MySQL à partir d'une BDD Excel. Pour cela je récupère un tutoriel sur OpenClassroom, ajoute une ligne à la BDD proposée avec le tutoriel afin d'avoir une colonne contenant des adresses mail. Je crée par la suite une base de données de test nommée **dbmlgasb05test** et y crée une table nommée **personnes** avec comme colonnes **.id, prenom, mail**.

J'y insère les données de la BDD Excel avec le fichier **.csv** que j'ai créé en exportant le fichier correspondant aux données.

### **La base de donnée test est remplie et fonctionne**

Par la suite, je dois me mettre au php, je recherche donc comment envoyer un mail en php, trouve la fonction **mail()** et la fait fonctionner correctement en rentrant les données souhaitée à la main dans un fichier nommé **envoiMail.php**. Par la suite, il me faut récupérer ces données dans ma BDD de test. Je dois donc m'y connecter et y envoyer une requête qui récupère les données souhaité. Je crée donc un fichier **connect\_base.php** dans lequel j'utilise la fonction **mysqli\_connect()** afin de me connecter à la base **dbmlgasb05test**. Une fois connectée, je dois récupérer les données souhaitée, à savoir le prénom et le mail de chaque personne, selon un critère. N'ayant pas de dates de naissances inscrite, je décide de récupérer les données de tous ceux dont l'**id** est strictement inférieur à 5.

Ces donnée récupérée, je les parcours une par une et les insères dans des variables que j'appelle par la suite dans ma fonction **mail()**. Tant que j'ai des données à parcourir, je crée et envois les mail personnalisés.

### **Le script me permet d'envoyer un mail personnalisé à plusieurs personnes, selon un critère que je choisi.**

Je dois désormais automatiser l'appel du script, après quelque que recherche, je trouve le système de Cron, qui permet d'appeler un script php quand je le souhaite sans m'en occuper. Après un essais non concluant avec cron-jobs.org, je décide d'héberger mes script sur Lixia.li qui fourni une BDD MySql et un hébergement gratuit en plus des tâche Cron tout aussi gratuites. J'y transfère donc mes script et ma BDD et tente une tâche Cron, cependant, une erreur système, en cours de réparation, m'empêche d'aller plus loin avec Cron et de le tester.

## **Semaine 2**

La deuxième semaine a été une semaine où j'ai été beaucoup bloquée. Car après avoir été notifié que mon problème de tâche Cron reporté précédemment, j'ai souhaité enfin essayer leur système. Malheureusement, la tâche Cron a beau se créer, elle ne fonctionne pas. J'ai donc immédiatement reporté le problème lundi et été en attente de réponse jusqu'à Jeudi matin. En effet, un membre du staff de Lixia m'a expliqué que ce problème provient d'un problème de sécurité entre le site et leur Cron. Ils devraient résoudre ce problème en quelques jour.

Entre temps j'ai pu commencer le backOffice du Club et finir de le faire fonctionner sur la BDD test. J'ai aussi pu voir Jeudi soir la salariée du Club, Marianne, afin de lui parler du problème rencontré avec le .csv que j'ai reçu précédemment.

## Le backOffice du GASB05

Le back Office est une application conçu avec QT qui sert actuellement à mettre à jour une BDD distante hébergée sur PHPMyAdmin avec un fichier .csv (ici récupéré d'un fichier Excel). Ce backOffice a pour but d'être amélioré afin de correspondre aux demandes du club et pour rassembler un maximum de fonctionnalité dans une même application.

Actuellement, elle permet d'éviter une double saisie.

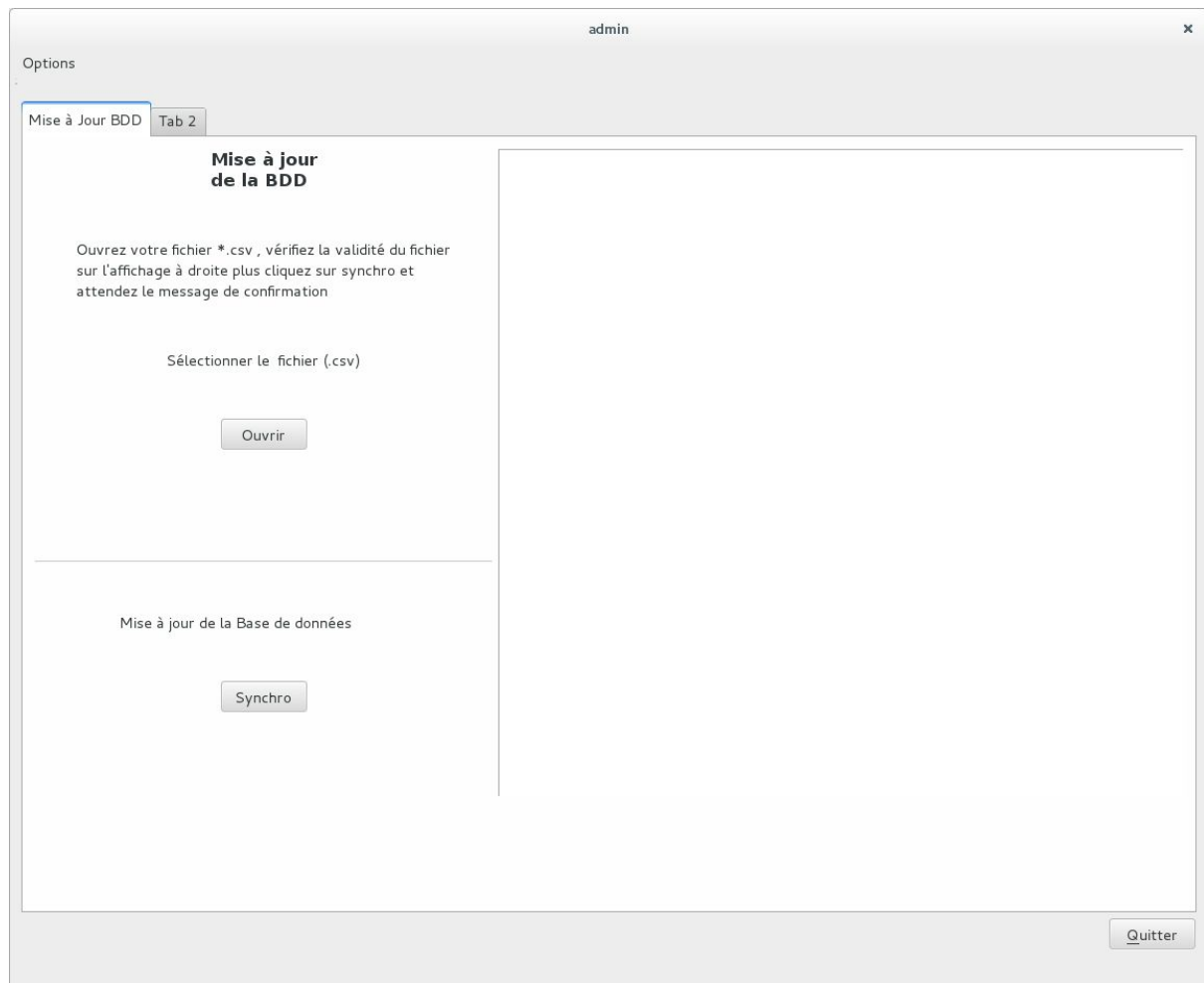
Affichage :

L'accès à l'application se fait avec une **fenêtre de connexion** :



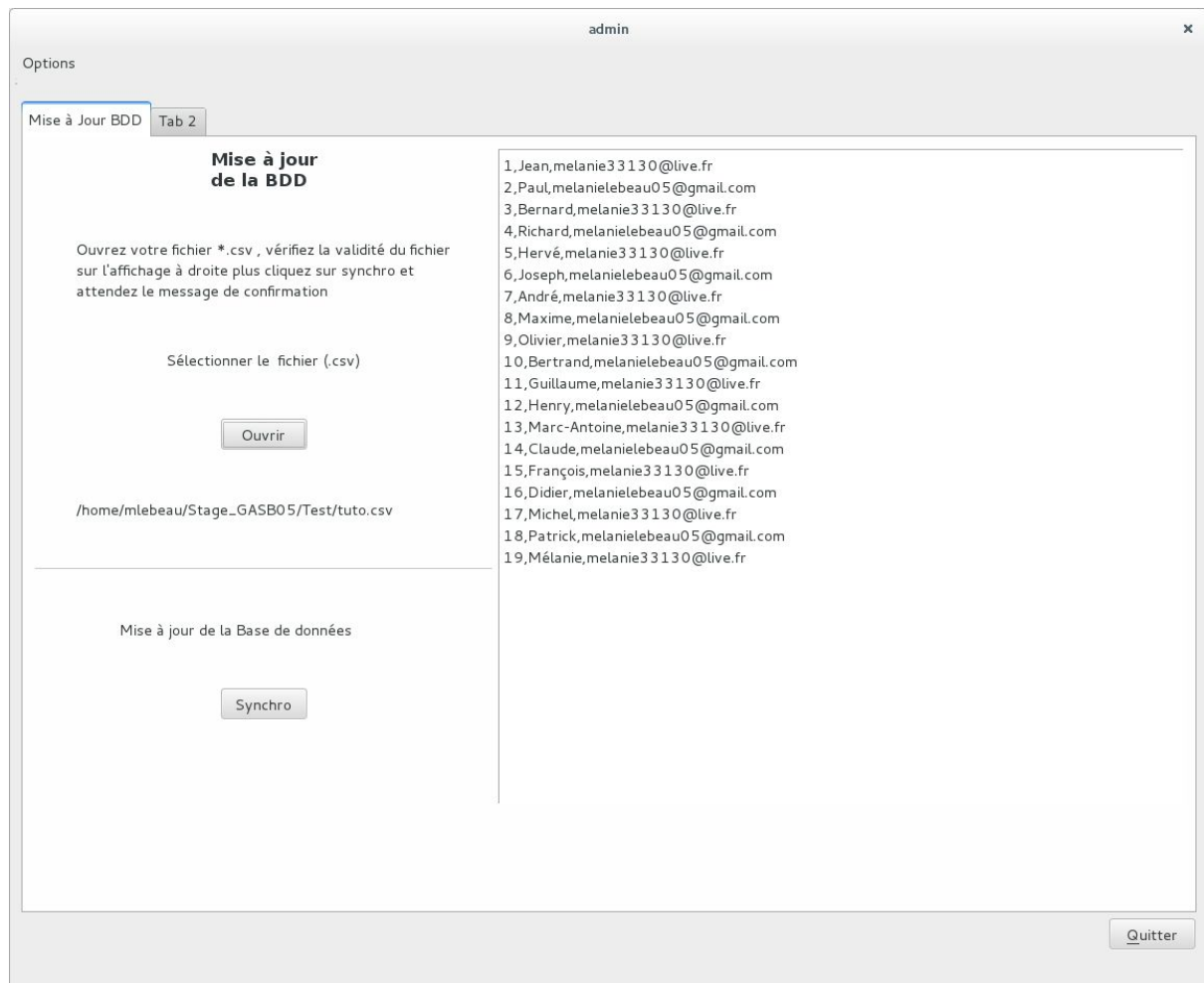
The image shows a Qt-style login window titled "Connexion". The window has a light gray background. At the top, the title "Connexion" is displayed in a bold, serif font. Below the title, there are two input fields. The first is labeled "Login" and contains the text "admin". The second is labeled "Password" and contains a series of dots. At the bottom of the window, there are two buttons. The left button is labeled "Connexion" and features a blue person icon. The right button is labeled "Quitter".

Puis l'application se présente actuellement avec un multi-onglet (Afin de rajouter d'autres usages au logiciel) mais seul un onglet (tab) a été actuellement crée :

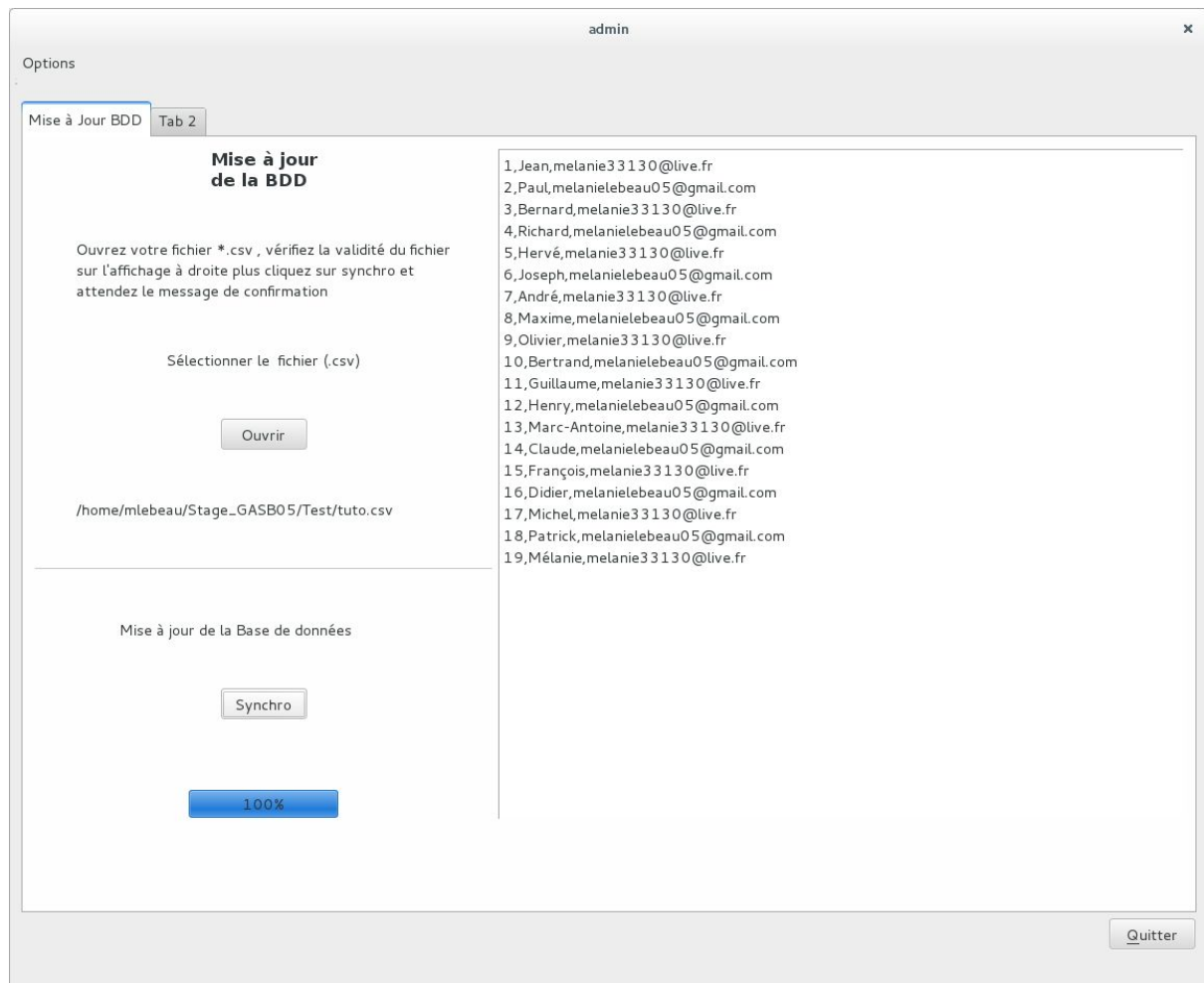


Sur cet onglet, il suffit d'ouvrir le fichier d'extension **.csv** avec le bouton **Ouvrir**, le panneau à droite permet de vérifier le contenu du fichier, puis le bouton **Synchro** permet la synchronisation avec la Base de donnée. Il n'est actuellement pas possible de choisir la base de donnée ni la table mais j'ai pour projet de programmer le bouton **Option** situé dans la barre de menu afin de rajouter cette fonctionnalité.

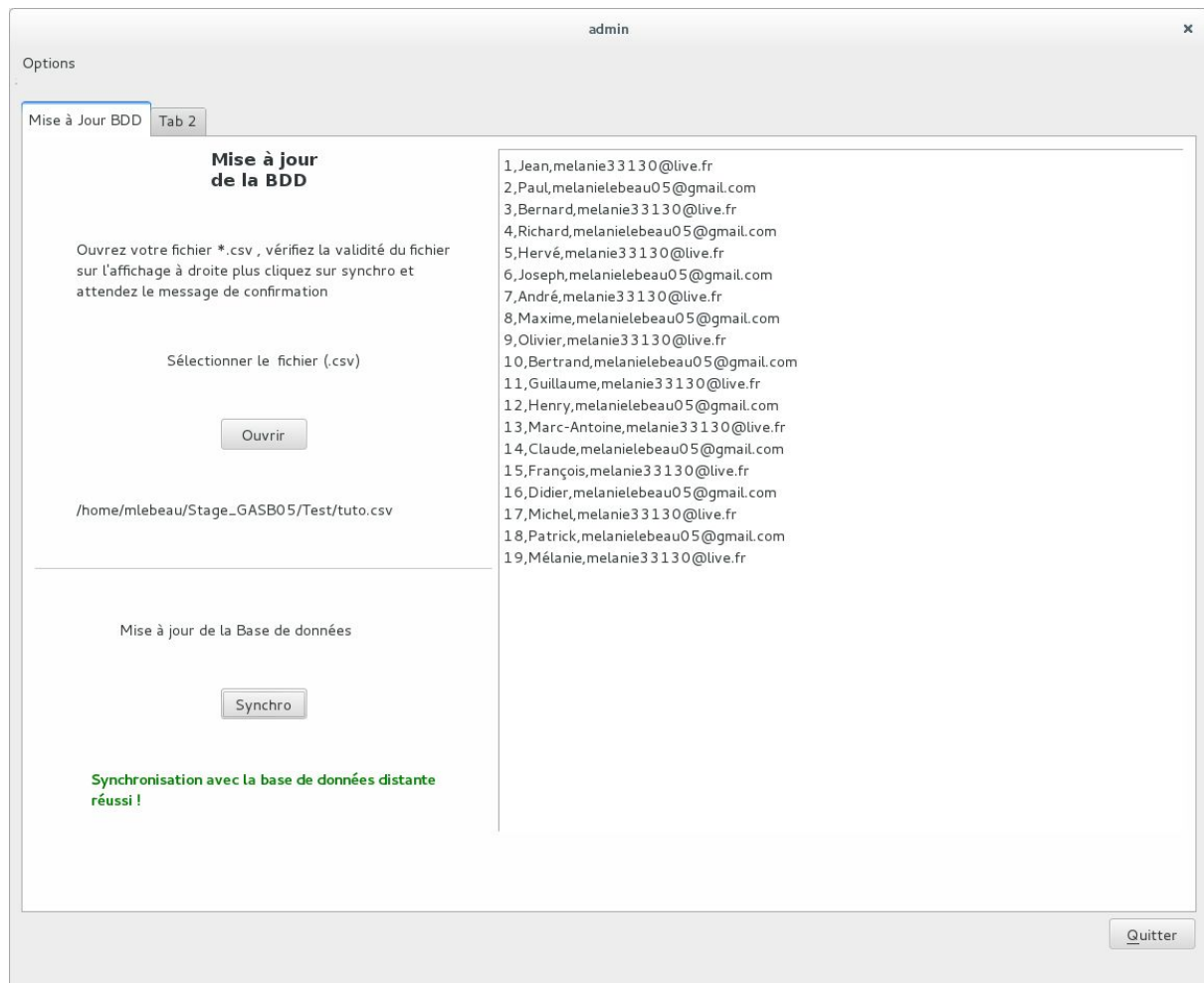
Le logiciel en fonctionnement :



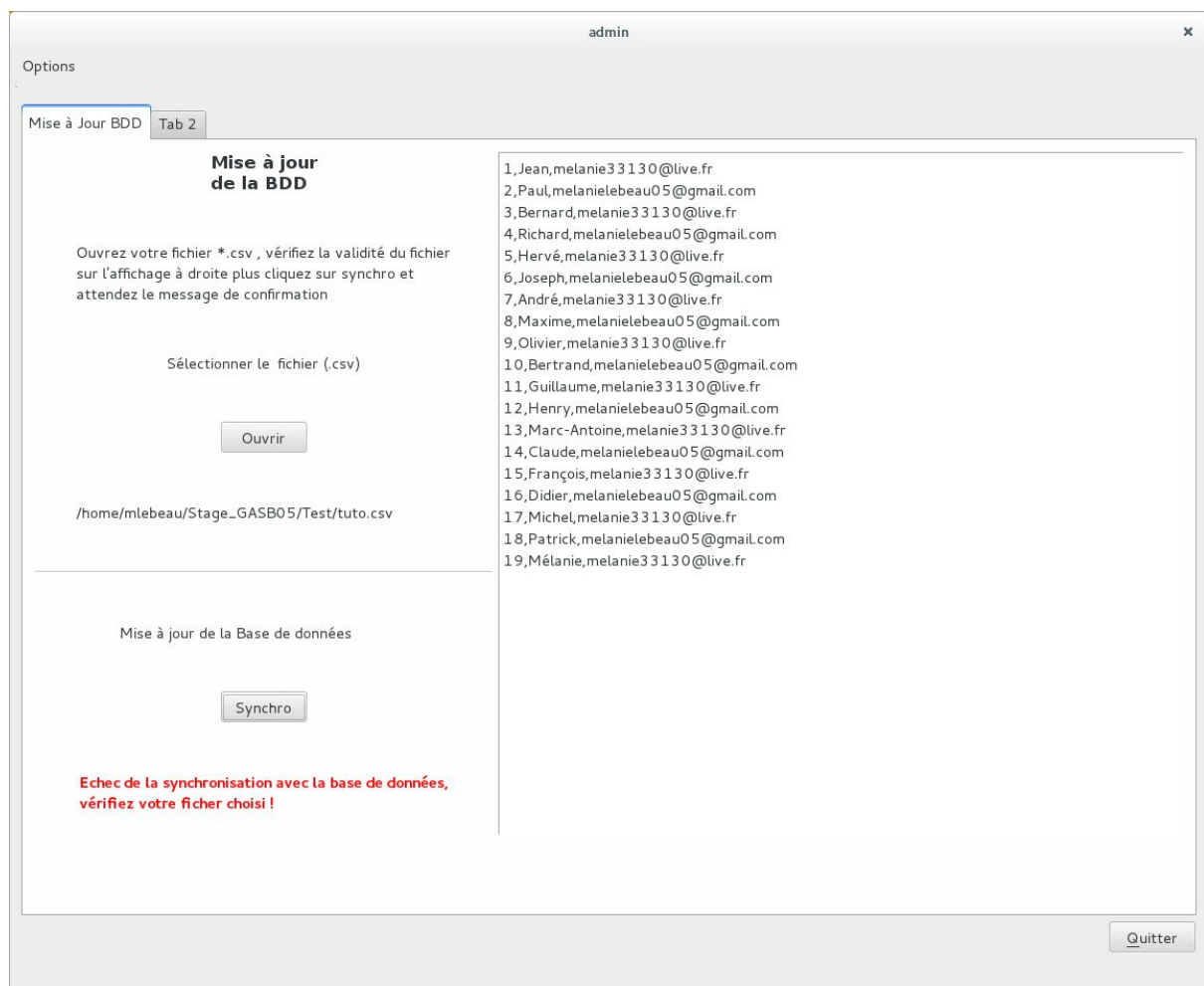
Le fichier est chargé, on peut le vérifier sur le panneau à droite.



Le bouton **Synchro** est cliqué, une barre de chargement apparait, elle progresse en même temps que la synchronisation.



La barre disparaît uniquement si la requête a bien été chargée (qu'elle fonctionne ou non), un message apparaît, ici pour dire que la synchronisation est réussie.



Ici pour dire que la synchronisation a échoué, et qu'il faut alors vérifier le document, afin de voir si l'erreur ne provient pas de lui.

En cliquant sur **Ouvrir** ou **Synchro** le message disparaît.

## Le problème rencontré avec le CSV

Le fichier que j'ai récupéré que Marianne m'a transmis est étrangement vide dans Excel tableau et ne contient que des dièses (#) en mode texte. D'après mes professeurs, l'erreur viendrait d'une mauvaise exportation.

Après avoir parlé avec Marianne, la salariée du club, il s'avère que l'erreur n'a pas l'air de venir de son exportation car elle m'a dit télécharger un fichier présent sur le site de la fédération. Je vais donc lui envoyer un message afin qu'elle me renvoie ce fichier et voir si cette fois-ci ce dernier est valide.



## Semaine 3

Durant la 3ème semaine, j'ai principalement travaillé sur la fenêtre **Option** qui me posait problème car elle ne mettait pas à jour la connexion de la base de données. Après avoir discuté avec mes professeurs, j'ai donc compris qu'en écrivant en "dur" (directement dans le code) les données de connexion, je ne pouvais pas mettre à jour la connexion. J'ai donc eu 2 choix: soit insérer les données de connexion dans une autre base de données, soit je créer un fichier texte d'une extension particulière qui ne sera lu que par l'application et qui est stockés dans l'application.

J'ai donc choisi la deuxième solution.

### Solution de création de fichier

Cette solution consiste à créer un fichier au premier lancement du logiciel puis à stocker ce fichier dans l'application.

Pour cela j'ai créée une fenêtre qui se lance si aucun fichier d'un nom précis n'existe dans l'application. Une fois cette fenêtre validée, si la connexion est réussie, je peux accéder alors à la fenêtre de connexion.

Si j'accède à la fenêtre d'option, alors je peux modifier ces données de connexion et si la connexion se fait, alors le fichier contenant ces données en le supprimant et en le re-écrivant avec ces nouvelles données.

**Fenêtre d'ajout de base de données:**

Dialog

## Ajout de la Base de Données

Hôte

UserName

PassWord

Show Password

Database

Table

Load Table

Valider

Quitter

### Fenêtre d'option:

Dialog

**Option Synchronisation  
Base de données**

Hôte

UserName

PassWord

Database

Table

### Fenêtre d'erreur dans la base de données :



## Semaine 4

Toujours en attente de réparation du Cron, je me suis penché cette semaine sur la "version Club". C'est à dire la modification du logiciel et la création de la base de données afin de l'adapter au club GASB05.

La base de données que j'utilise est toujours la mienne car j'utilise mon compte afin de faire mes tests, j'appelle donc cette version la **version bêta** . Une fois que le logiciel sera opérationnel en version club je créerais aussi le compte Lixia.li du club.

### La base de données du club version bêta

Afin de créer la version bêta de la base de donnée du club, j'ai récupéré le fichier .csv contenant les données souhaitée auprès de Marianne, la salariée du club. J'ai par la suite crée un dictionnaire des données comme vu en cours en adaptant les noms de données à la base de données (par exemple le champs "nom" du fichier est devenu "licNom" dans la BDD).

Ainsi, le dictionnaire des données ressemble à cela:

Nom dans le fichier .csv	Nom dans la Base de données	Type	Commentaire
cd_org	orgCd	AN	
lb_org	orgLib	AN	
id_lice	licId	N	
nom	licNom	AN	
prenom	licPrenom	AN	
adresse	licAdresse	AN	Rue + Numéro dans la rue
code_postal	licCdePostal	AN	Numérique avec 0 au début
lb_cmne	cmneLib	AN	
cd_lic	licCd	AN	
qualification	licQualif	DATE	
sexe	licSexe	AN	
taille	licTaille	N	Taille en cm
cd_cat	catCd	AN	

cd_ass	assCd	AN	Valeur : A ou A+ ou B ou B+ ou N
naissance	licDteNaiss	DATE	
numero_licence	licNum	AN	
certificat_medical	licCertifMed	DATE	
nationalite	licNationalite	AN	
telephone_domicile	licTelDom	AN	Numérique avec 0 au début (mais non inclu dans le fichier)
telephone_professionnel	licTelPro	AN	Numérique avec 0 au début (mais non inclu dans le fichier)
telephone_portable	licTelPort	AN	Numérique avec 0 au début (mais non inclu dans le fichier)
mail	licMail	AN	
recevoir_offre_commerciale	licRecOffreComm	AN	Valeur : Accepte/Refuse
recevoir_non_commerciale	licRecNonComm	AN	Valeur : Accepte/Refuse
Textbox1	clubNbrLic	AN	Nombre total de licence (avec texte)

J'ai aussi écrit le script de création de la table:

```

/* Présentation :
nomScript: gasb05BDD_base.sql
nomBase: GASB05BDD
Auteur: Mélanie Lebeau
Date de création : 30/01/2018
*/

```

```

/*****/
/*Suppression des tables existantes*/
DROP TABLE IF EXISTS Licence;

/*****/
/*Création de la table Licence */

```

```

CREATE TABLE Licence(
    orgCd VARCHAR(7),
    orgLib VARCHAR(50),

```

```

        licId INT,
        licNom VARCHAR(50),
        licPrenom VARCHAR(20),
        licAdresse VARCHAR(100),
        licCdePostal VARCHAR(5),
        cmneLib VARCHAR(50),
        licCd VARCHAR(10),
        licQualif DATE,
        licSexe VARCHAR(1),
        licTaille INT,
        catCd VARCHAR(3),
        assCd VARCHAR(2),
        licDteNaiss DATE,
        licNum VARCHAR(8),
        licCertifMed DATE,
        licNationalite VARCHAR(20),
        licTelDom VARCHAR(10),
        licTelPro VARCHAR(10),
        licTelPort VARCHAR(10),
        licMail VARCHAR(100),
        licRecOffreComm VARCHAR(7),
        licRecNonComm VARCHAR(7),
        clubNbrLic VARCHAR(35)
)ENGINE=InnoDB CHARSET=utf8;

```

```

/*****

```

```

/*Création de la primary key*/
ALTER TABLE Licence
ADD PRIMARY KEY (licNum);

```

J'ai par la suite créée un table dans ma BDD nommée **Licence** et dans laquelle j'ai importé ce script puis le fichier .csv.

En regardant la BDD par la suite j'ai pu remarqué que les champs vides sont bien restés vide mais que les données de type **DATE** ne se sont pas inscrite. Car la données du fichier est sous forme **dd/mm/yyyy** tandis que dans une base de données elle s'écrit sous forme **yyyy-mm-dd**. Ainsi, soit je passe ce champs sous forme **VARCHAR** et je fait le traitement dans mon script php, soit je fais le traitement du champs dans le logiciel qui mets à jour la BDD.

Après avoir discuté avec M.Thomassin, je dois modifier le logiciel afin de réaliser la seconde partie, (malgré mes craintes de ralentissement du logiciel.).

Cependant, je dois analyser le contenu de la valeur que je parcours afin de vérifier si elle est au format **dd/mm/yyyy** et si c'est le cas, je devrais spliter sur le "/" afin de réorganiser le champs sous forme **yyyy-mm-dd** et l'insérer dans la requête. Si ce n'est pas le cas, j'ajoute simplement la données à ma requête sans traitement. Je me suis donc penché sur le premier problème : Comment analyser le "format" d'un string?

Après de nombreuses recherche, j'ai trouvé que l'utilisation des **regular expression** (expression régulière) aussi appelée **Regex** conviendra parfaitement à ce que je souhaite faire. Après de nombreuse recherche sur la manière de fonctionnement des RegEx, j'ai modifié et testé mon code. Le format est transformé et inséré dans ma BDD sans problème.

## Semaine 5

Toujours en attente de réparation du Cron, j'ai avancé cette semaine avec la version club. J'ai rencontré quelques problèmes avec les données du fichier .csv. En effet, certaines données contenaient une virgule et était entouré de double quotes, une ou plusieurs doubles quotes ou une apostrophe.

Cela posait problème car dans mon code je "split" sur les virgules (sépare le fichier sur les virgules) et je me retrouvais alors avec plus de données que nécessaire, et qui se retrouvaient dans la mauvaise donnée. J'ai reçu de l'aide de M. Thomassin pour résoudre ce problème. La solution consistait à insérer les éléments dans une liste en vérifiant leur contenu. Si la donnée contenait deux doubles quotes, alors je les remplace et j'ajoute la donnée suivante à celle actuelle.

De plus, l'apostrophe posait problème car j'entoure les données de ma requête par des apostrophe. Ainsi, la fin de ma données n'était pas "insérable" dans ma BDD. J'ai simplement dûs échapper les apostrophe en les doublant.

Pour les doubles quotes, j'ai reçu aussi l'aide de M.Thomassin . Si ma donnée contenait 2 doubles quotes à la suite, alors je les remplace par rien.

## Semaine 6

Durant la dernière semaine de stage, j'ai pu créer le compte version club ainsi que la base de donnée. J'ai donc dû modifier le logiciel afin qu'il marche parfaitement avec ce nouveau compte. J'ai aussi rajouté une fenêtre en cas d'erreur dans la première saisie des données la base de donnée. Puis j'ai réglé des petit détails visuel (désactivation des pushButton principalement) puis j'ai fini par réussir à programmer la Progress Bar.

