1. **Se connecter :**

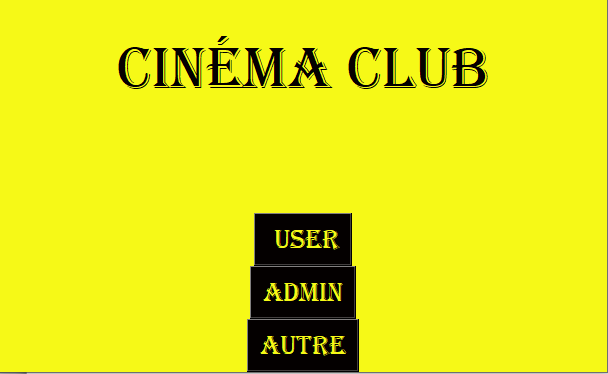
Cette page, est une page d’accueil pour se connecter à l’application, quand l’utilisateur appuie sur le bouton « **se connecter** », l’utilisateur passe à la prochaine étape **Login**, dans le code ca renvoi a la classe  **Interf\_connexion**



1. **Login :**

**2.1 Choix du profil :**

Cette page permet à l’utilisateur de l’application de choisir son profil, pour se connecter, dans notre cas, on’ a 3 types de profils (ADMIN, USER, AUTRE). Cette interface correspond dans le code a la classe **Interf\_connexion**



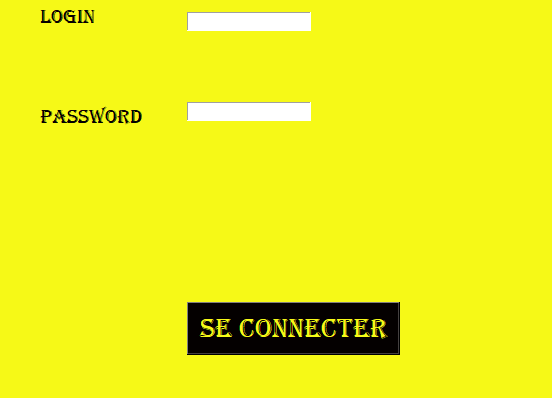
* 1. **S’identifier :**

Cette page sert pour que l’utilisateur entre ces cordonnées **Login** et **password,** cette page s’affiche juste pour les deux profils **USER** et **ADMIN**, **AUTRE** n’a pas besoin de s’identifier.

Les informations de toutes les employées sont dans la base de données « **myDB.db** » exactement dans la table **« Empoyé ».**

Pour Valider les cordonnés il faut appuyer sur le bouton **« Seconnecter ».**

La gestion et la vérification des données est faite dans la **classe Login.**



* 1. **Connexion :**

En fonction de profil de l’utilisateur une interface s’affiche, selon les droits de chaque profil.

* **Admin : (**login : admin, mot de passe : admin ’dans table Empoyé’)

Il a 3 choix, **Gestion de L’outil, Gestion De PLANNING** et **Gestion de BILLETERIE.**



* **User :**

Le profil **User** ne peut que Modifier les évènements assignés à lui et consulter les évènements présents.



* **Autre :**

Le profil **Autre** il peut que consulter les évènements.



1. **Fonctionnalités de l’admin:**

La fenêtre de l’admin dans le code est la classe **Accueil\_admin.**

**3.1 Gestion de l’outil :**

Quand l’admin choisi Gestion de l’outil cette interface s’affiche pour choisir une action :

Dans le code cette interface est construite dans la classe **Outil**

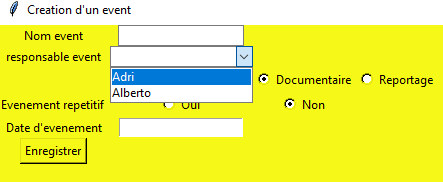


* **Ajouter un évènement :**

Seul l’admin peut ajouter un évènement, pour cela il suffit de cliquer sur lebouton **« AJOUTER UN EVENEMENT »,** cette partie est gérer dans le code par la classe **Creation\_Event**.

L’admin doit remplir, ‘**Nom event’**, et choisir **‘responsable event’,** le responsable de l’évènement peut prendre plusieurs évènements tant que dans la base de donné la variable **Available=1**, dans notre cas je propose que 2 users.Ensuite il faut remplir tous les autres données, et appuyer sur **enregistrer**. Cette action crée une ligne de donné dans la base de données.

Quand l’évènement est créé, il a le statut « crée » avec toutes les données remplis dans la base de données.



* **Supprimer un Evènement :**

Seul l’admin peut supprimer un évènement, pour cela il suffit de cliquer sur lebouton **« SUPPRIMER UN EVENEMENT »,** cette partie est gérer dans le code par la classe **Supprimer\_Event**, ensuite choisir le nom de l’évènement à supprimer.



* **Consulter les évènements :**

Tous les profils (**ADMIN, USER, AUTRE**) ont le droit de consulter les évènements créé.

Cette partie est géré dans le code par la **classe Consulter\_Event**



* 1. **Gestion de planning :**

Affiche les congés de différentes employées :



* 1. **Gestion de Billetterie :**

Seul l’admin a le droit de voir l’état de la billetterie, le prix est fixé à 30 EUR par billet, dans la base de donné. Cette interface affiche juste les évènements avec le statu Terminé.

Cette partie est gérée par **la classe Billetrie.**



1. **Users :**

L’User a le droit juste de modifier et consulter les évènements :

**4.1 Modifier un évènement :**

Quand l’utilisateur remplit tous les information le statut passe a terminer sinon il est en cours .

