GUI

# Fenêtre connexion/enregistrement d’un compte utilisateur

## Vue

Pour définir notre interface graphique nous utilisons le format de fichier FXML. Le FXML est le format utilisé par javaFX pour définir des interfaces graphiques. Il utilise tout simplement la syntaxe XML

Pour cette vue on utilise le fichier LoginRegister.fxml. Cette vue contiendra plusieurs éléments. Le conteneur principal de notre vue sera un « Pane ». Ce conteneur est la classe de base pour les panneaux de mise en page. Il permet d’y insérer des éléments graphiques (conteneur, contrôle, composant, etc.) et de les positionner librement. Dès que l’on ajoute des éléments dans un conteneur nous les appelons les enfants et dans notre conteneur principal nous avons trois enfants qui seront :

* Un label qui sera le titre de notre vue
* Un GridPane pour l’espace de connexion
* Un GridPane pour l’espace d’enregistrement

On utilise des GridPane pour les espace de connexion et d’enregistrement car il permet d’y ajouter des enfants dans une grille flexible de lignes et de colonnes. Un enfant peut être placé n’importe où dans la grille et peut s’étendre sur plusieurs lignes/colonnes. Cela nous permet de faciliter l’alignement de nos composants que l’on ajoute dans nos GridPane. Les composant utilisé dans nos GridPane pour la récupération d’une entrée utilisateur (email, nom, prénom, mot de passe) se font avec des Textfield et Passwordfield pour les mots de passe. Et pour ce qui est des contrôles on utilise des boutons.

## Contrôleur

Dans notre fichier Controller\_loginRegister.java on aura deux événements qui portent sur les deux boutons de l’espace de connexion et d’enregistrement. Chacun de ces événements appelleront une méthode de la BL (Business Logic).

Un évènement, clickLoginButton, appellera la méthode login qui vérifiera si l’email est bien dans la base de données et que le mot de passe correspond bien à cet email. Si ce n’est pas le cas, la méthode nous retournera une erreur. On traite cette erreur par l’affichage d’un texte en rouge « Identifiant ou mot de passe incorrect ».

Un autre événement, clickRegisterButton, appellera la méthode register qui vérifiera si l’email est déjà dans la base de données. Si l’email est dans la base de données cette méthode retournera une erreur. SI la confirmation du mot de passe n’est pas identique nous afficherons aussi une erreur. On traite ces erreurs par le soulignement en rouge des champs qui sont faux.



[Capture avec les cas d’erreur sur le login]

[Capture avec les cas d’erreur sur le register]

# Fenêtre compte bancaire

## Vue principale

### Vue

Pour cette vue nous utilisons le fichier bankAccount.fxml. Cette vue aura comme conteneur principal un AnchorPane. Ce conteneur nous permet de fixer les bords des enfants à une distance par rapport au bord de notre conteneur. Cela est pratique lorsque l’on redimensionne notre fenêtre.

Dans ce conteneur nous avons deux enfants :

* Un FlowPane pour lister les comptes bancaires
* Un bouton pour la création d’un compte bancaire

Nous utilisons un FlowPane pour lister les comptes bancaires car il nous permet, lorsque l’on ajoute un enfant, de ne pas devoir spécifier la position notre enfant dans le conteneur. Il se place automatique, en haut à gauche (par défaut), les uns après les autres sur une seule ligne. Et une fois arrivé au bord de notre conteneur cela recommence sur une seconde ligne. Lorsque l’on redimensionne notre fenêtre, les enfants qui sont dans notre FlowPane se repositionneront de tel sort à en avoir un maximum sur une ligne (voir image ). Pour ce qui est du contrôle pour l’ajoute d’un compte bancaire, nous utiliserons un bouton.

### Contrôleur

Notre classe Controller\_bankAccount implémentera l’interface Initializable qui nous permet d’avoir la méthode initialise. Cette méthode nous permet, lorsque notre classe est appelé, d’initialisé notre contrôleur. Dans notre cas nous récupèrerons la liste des comptes via la méthode getBankAccounts de la classe bankAccount. Pour chacun de ces comptes, nous créerons un AccountDisplayer que nous ajouterons à notre FlowPane.

De plus nous ajoutant une action à notre bouton, createBankAccount\_button, qui appellera la méthode clickCreateBankAccountButton. Cette méthode chargera la vue de création d’un compte bancaire. Lorsque l’utilisateur aura remplis tous les champs et aura validé sa création, nous récupérons ces données et nous créerons une instance de la classe BankAccount. Ensuite nous faisons le même procédé que pour l’initialisation. Nous créons un AccountDisplayer avec l’instance de notre classe BankAccount que nous ajoutons à notre conteneur.

Un AccountDisplayer est un GridPane qui contiendra le nom attribué à notre compte bancaire, le montant de notre compte bancaire ainsi que la date de notre dernière transaction. (à voir si on le met)



## Vue détaillée

## Vue création/modification