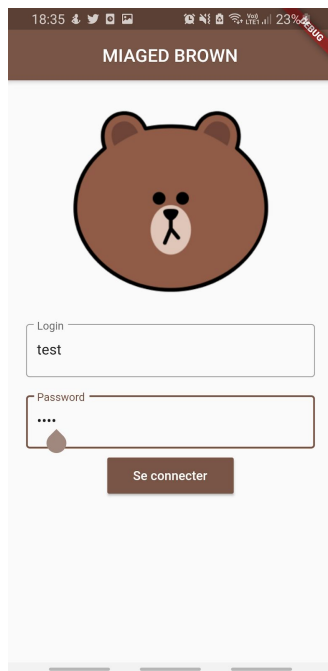


**ATTENTION les images utilisées appartiennent toutes à Line, elles ne sont pas libres de droits donc elles ne m'appartiennent pas. J'aurais aimé les faire moi même pour cette démo, mais mon matériel n'est pas fonctionnel.**

#### US#1 : [MVP] Interface de login :



Cette page apparaît après le splashscreen. Elle respecte tous les critères de l'User Story.

Cette page est réalisé par le fichier **"LoginPage.dart"**

C'est dans cette page que les informations de l'utilisateur vont être récupérées et stockées dans **AppData.dart**



#### US#2 : [MVP] Liste de vêtements + US#6 : Filtrer sur la liste des vêtements:



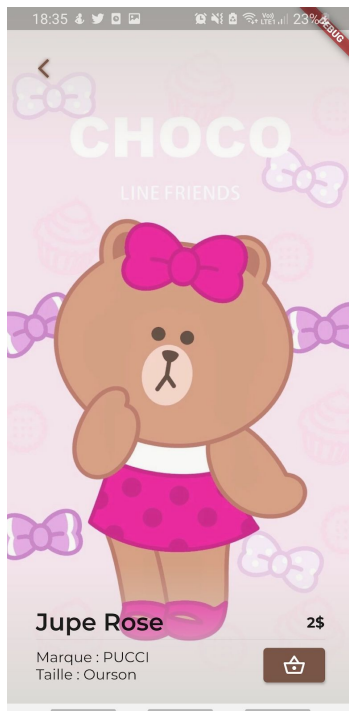
Lorsque l'utilisateur se connecte, il tombe sur cette page. La page respecte tous les critères des User Story.

**Si l'utilisateur clique sur l'article il est redirigé vers le détails de celui-ci.** (Je précise car je n'ai pas fait apparaître de bouton, le card en lui même est un bouton. )

Cette page est générée par ces fichiers :

- **MainScreen.dart** : Gère la BottomBar et donc la liaison avec le panier et la page de profil
- **BoutiquePage.dart** : Va afficher une liste de Cards (chaque article est un card) depuis Firebase en fonction de la catégorie sélectionnée par la TabBar ("All" est affiché par défaut).

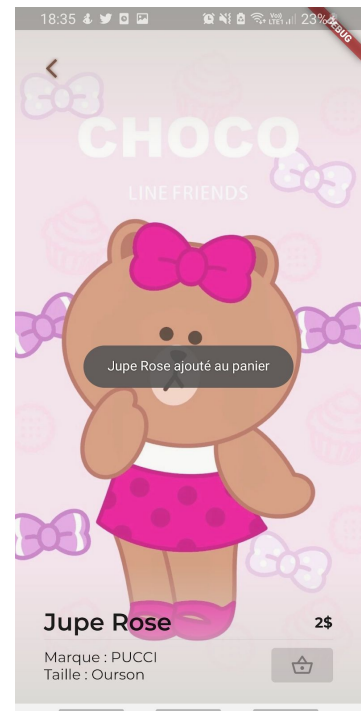
### US#3 : [MVP] Détail d'un vêtement :



Cette page est générée par **DetailsPage.dart** après que l'utilisateur ait appuyé sur un article pour voir les détails. Elle respecte tous les critères de l'User Story.

Dans le code, **BoutiquePage.dart** envoie à la class appData l'id Firebase de l'article.

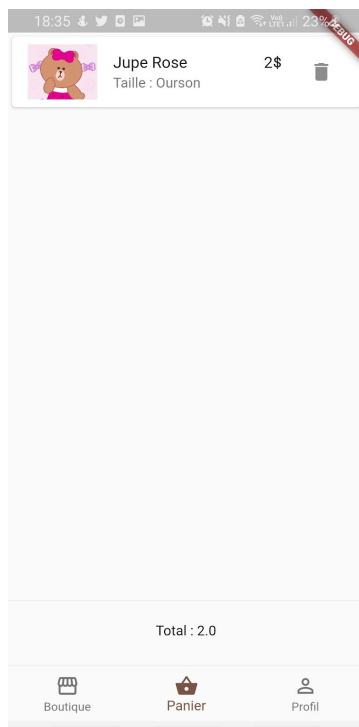
**DetailsPage.dart** va récupérer l'id de l'article via AppData et va ensuite chercher les informations sur Firebase à l'aide de cet Id. Lorsque l'utilisateur clique sur le bouton ajouter, l'article est ajouté dans la panier de l'utilisateur sur Firebase.



#### Ajout :

- Un message s'affiche lorsque l'utilisateur ajoute l'article au panier.
- Le bouton devient inutilisable après l'ajout de l'article, car il n'est autorisé de n'en mettre qu'un seul dans son panier.

### 3.4. US#4 : [MVP] Le panier :



Si l'utilisateur clique sur le panier figurant sur la BottomBar, il peut voir la liste des articles qu'il a dans son panier et son total. La page respecte tous les critères de l'User Story.

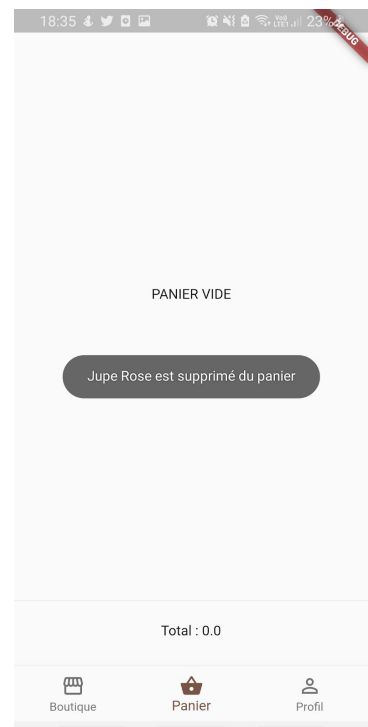
Cette page est gérée par **PanierPage.dart** et récupère la liste des articles du panier sur Firebase.

Pour le total, j'ai été obligée de passer par une class Produit.dart.

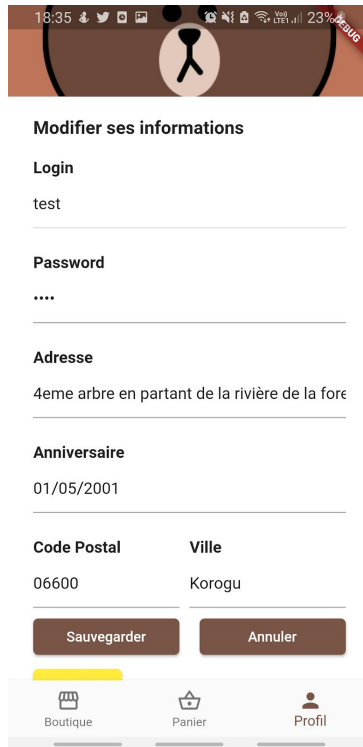
#### Ajout :

- Un message s'affiche lorsque l'utilisateur supprime l'article au panier.

Bien entendu, la suppression se réalise aussi sur Firebase.



### 3.5. US#5 : [MVP] Profil utilisateur :



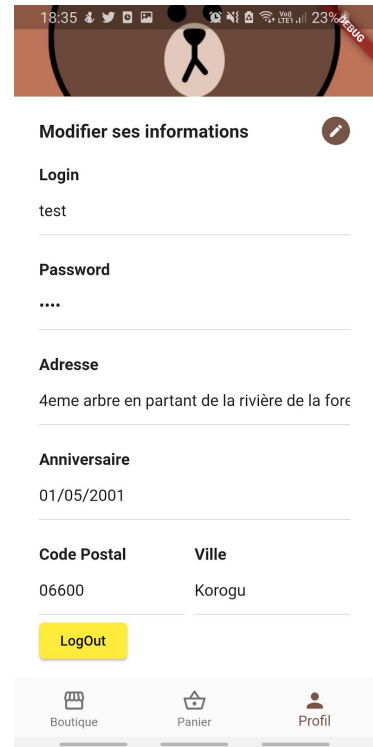
Si l'utilisateur clique sur le profil figurant sur la BottomBar, il peut avoir accès à ses informations. La page respecte tous les critères de l'User Story.

Une partie du code qui se charge de l'aspect visuel a été récupéré d'un code libre service disponible sur un github, puis je l'ai personnalisé.

Le code qui se garde de cette page est : **"Profilepage.dart"**.

#### Ajout :

- Pour modifier le profil, l'utilisateur doit cliquer sur l'icône en haut à droite.
- On peut annuler ses modifications.



#### Autre ajout :

SplashScreen (SplashScreen.dart) animé et un icon à l'application.

