

Développement Mobile

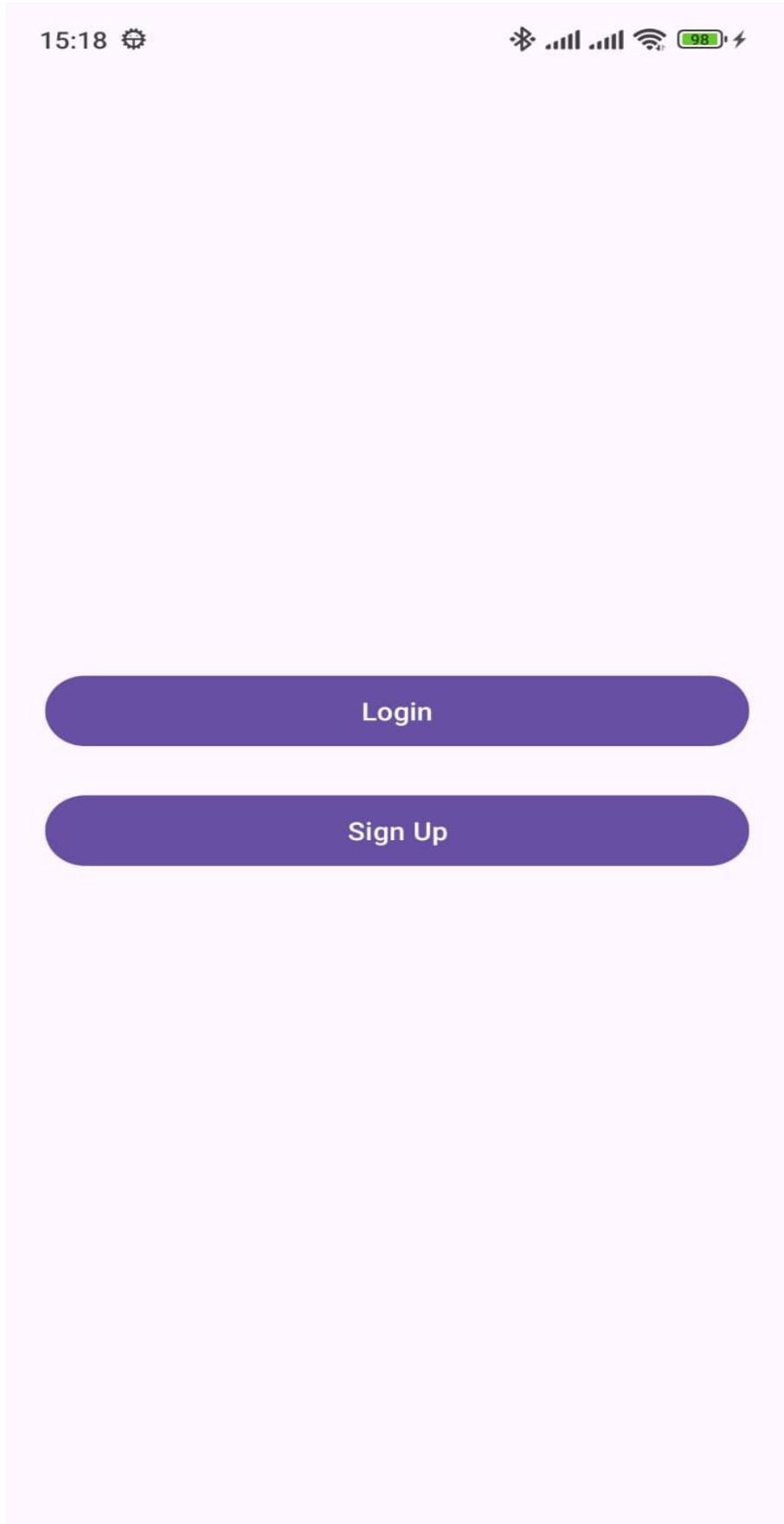
TP 9

12 fevrier 2026

2025/2026

I. Interface d'accueil :

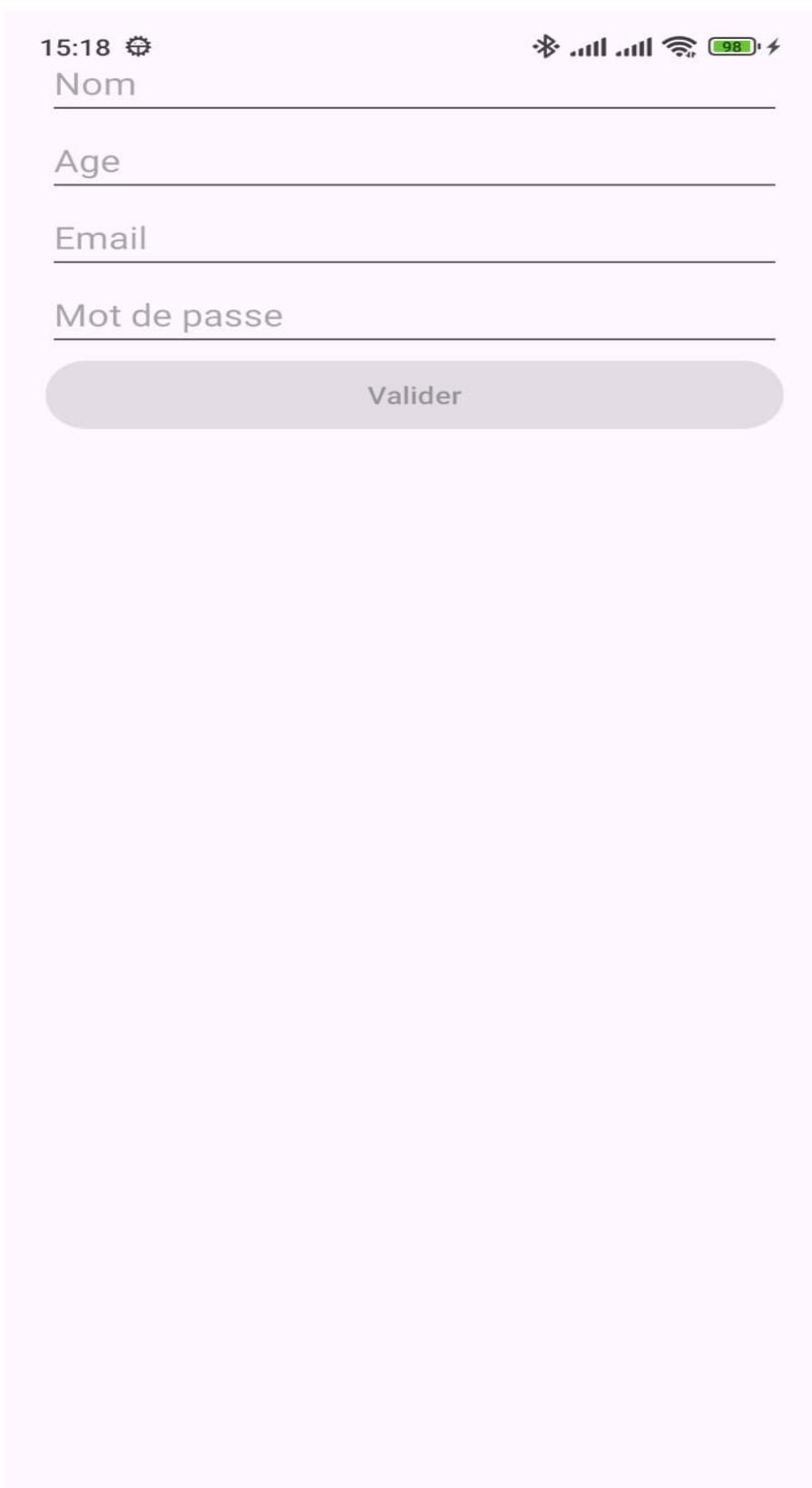
Cet écran représente la page d'accueil de l'application. Il contient deux boutons *Login* et *Sign Up* permettant à l'utilisateur de choisir entre la connexion à un compte existant ou la création d'un nouveau compte.



Interface d'inscription :

Écran initial (bouton désactivé) :

À l'ouverture de l'interface d'inscription, tous les champs sont vides et le bouton *Valider* est désactivé afin d'empêcher toute soumission tant que le formulaire n'est pas correctement rempli.



Erreur affichée si champ vide :

Lorsque l'utilisateur tente de valider le formulaire avec un ou plusieurs champs vides, un message d'erreur s'affiche sous le champ concerné pour indiquer que tous les champs sont obligatoires.

A screenshot of a smartphone screen displaying a form with three fields: Nom, Email, and Mot de passe. The 'Nom' field contains the text '20' and has a red border, indicating it is invalid. A red arrow points from the top right towards this field. To the right of the 'Nom' field is a black button with white text that says 'Nom obligatoire'. The 'Email' and 'Mot de passe' fields are empty and have no red borders. Below the form is a large grey button labeled 'Valider'.

15:19

Nom

20

Email

Mot de passe

Valider

Nom obligatoire

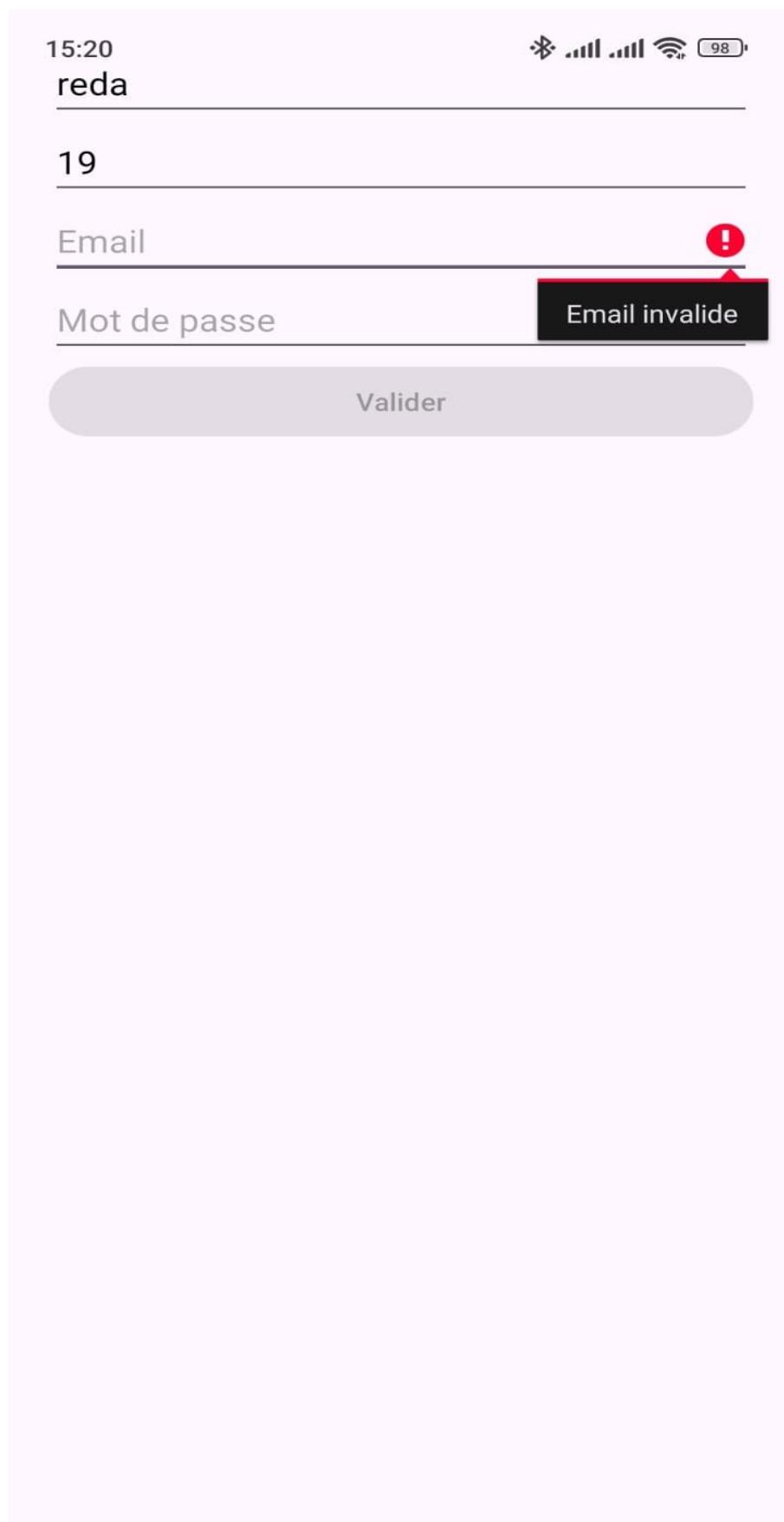
Erreur si âge invalide :

Si l'âge saisi n'est pas numérique ou s'il est négatif ou supérieur à 120, une erreur est affichée afin d'informer l'utilisateur que la valeur de l'âge est invalide.

The screenshot shows a mobile application interface with a light pink background. At the top, there is a status bar with the time '15:20', the user name 'reda', signal strength icons, and a battery level at 98%. Below the status bar, there are three input fields: 'Age' (which contains the placeholder text 'Age'), 'Email' (containing 'Email'), and 'Mot de passe' (containing 'Mot de passe'). To the right of the 'Age' field is a red rectangular button with the text 'Age obligatoire' and a white exclamation mark icon inside a circle. A blue arrow points from the text 'Age obligatoire' towards the 'Age' input field. At the bottom of the screen is a large, rounded rectangular button labeled 'Valider'.

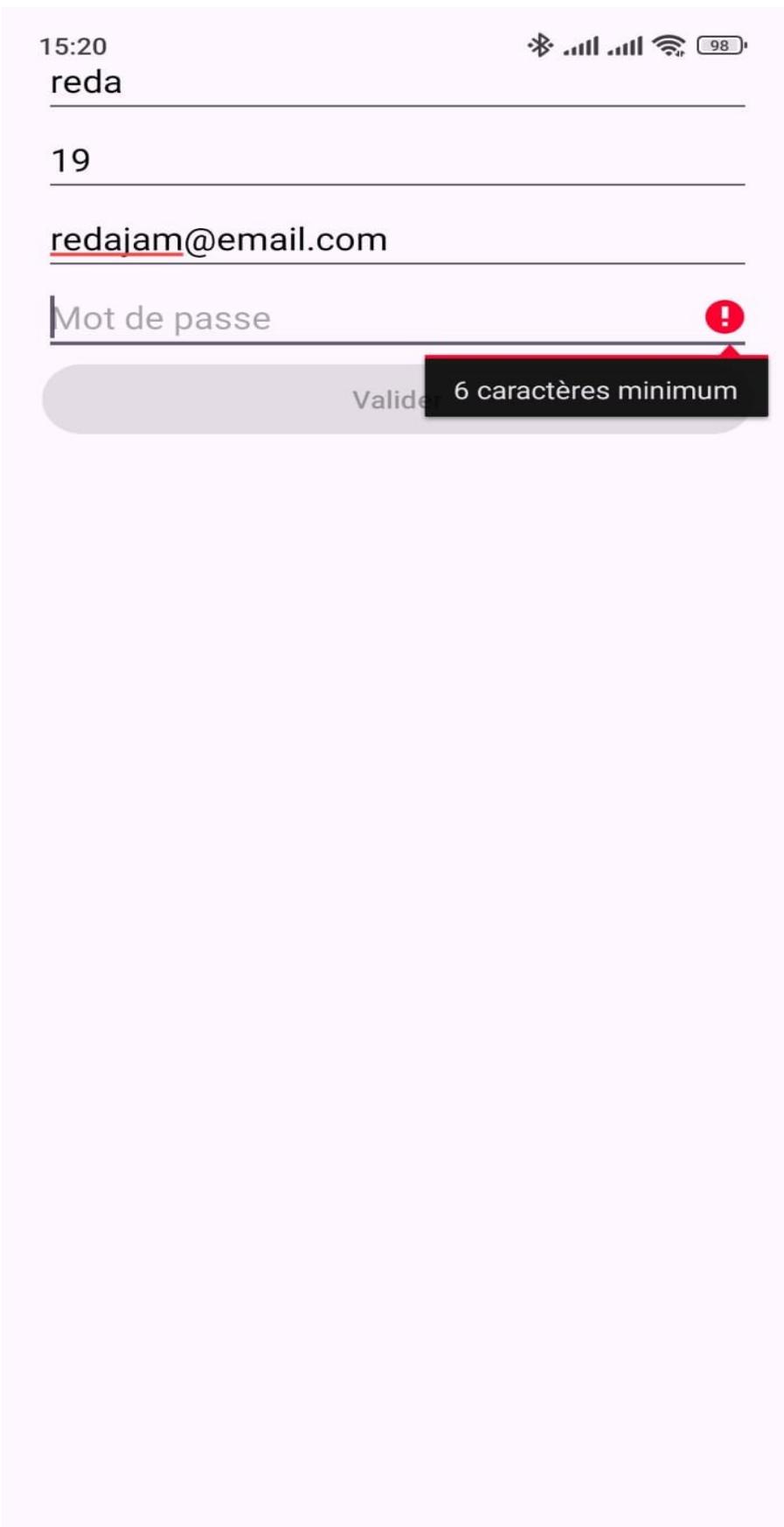
Erreur si email incorrect :

L'application vérifie que l'adresse email contient le caractère « @ ». En cas d'email invalide, un message d'erreur s'affiche pour demander une adresse correcte.



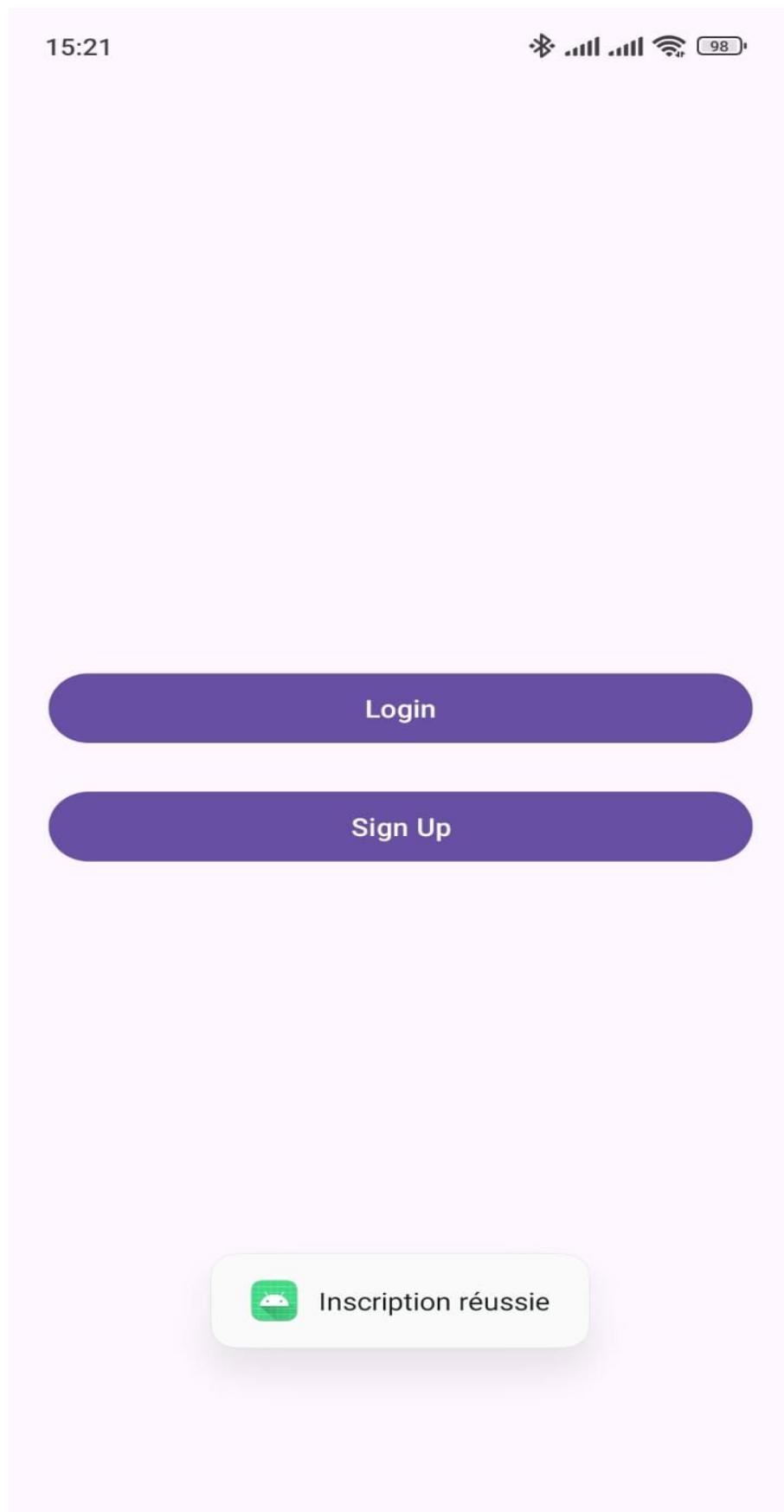
Erreur si mot de passe trop court :

Lorsque le mot de passe contient moins de 6 caractères, une erreur est affichée pour renforcer la sécurité et obliger l'utilisateur à saisir un mot de passe valide.



Capture après validation réussie (Toast affiché) :

Après la validation correcte de tous les champs, l'inscription est enregistrée dans la base de données et un message *Toast* confirme la réussite de l'opération.



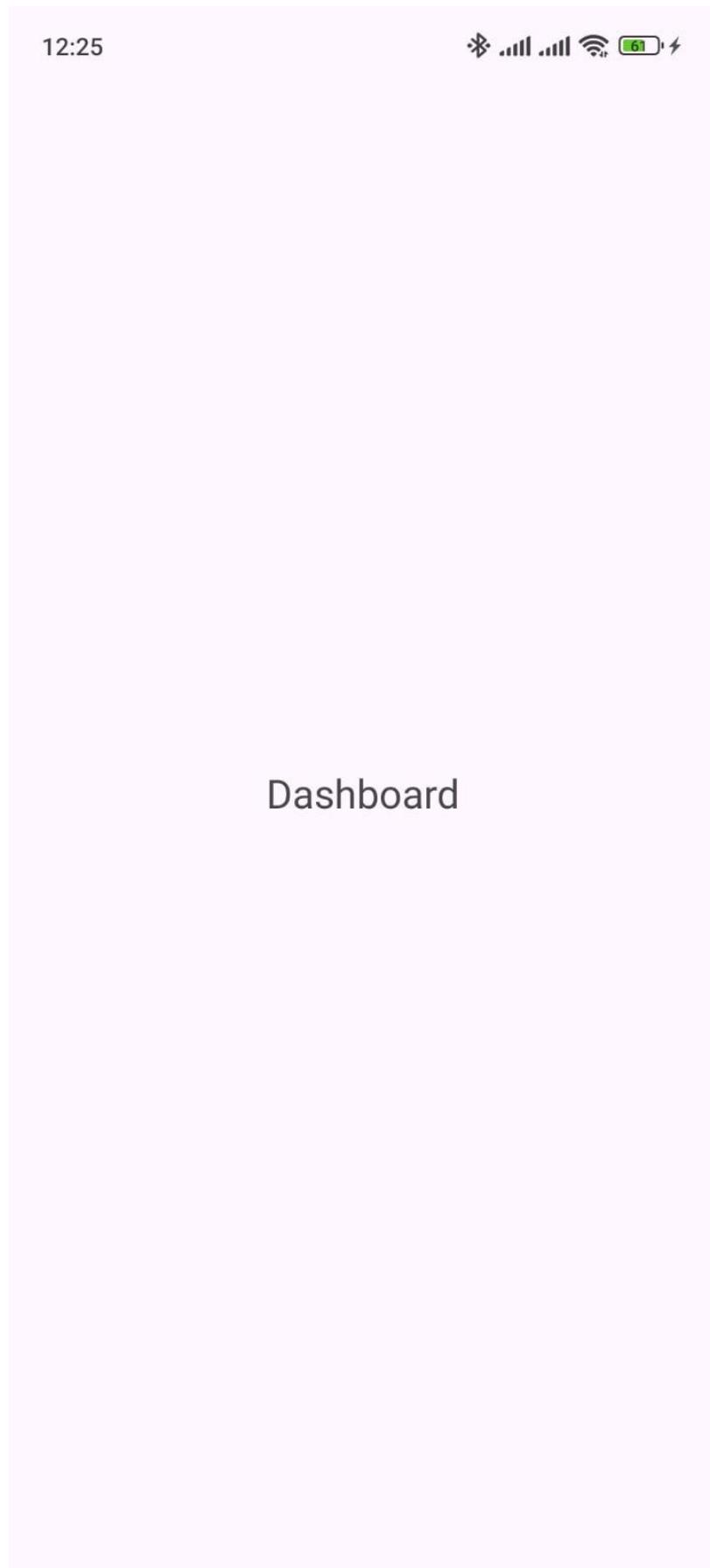
Interface de login – mot de passe incorrect :

Cet écran montre une tentative de connexion avec un mot de passe incorrect. L'application affiche un message d'erreur indiquant que les informations saisies ne correspondent pas aux données enregistrées.



Interface du tableau de bord (Dashboard) :

Cette interface correspond au tableau de bord accessible uniquement après une connexion réussie. Elle confirme que l'utilisateur est authentifié et a accès aux fonctionnalités principales de l'application.



Les réponses aux questions :

Pourquoi utilise-t-on `try / catch` ?

Réponse :

On utilise `try / catch` pour gérer les erreurs qui peuvent survenir pendant l'exécution du programme (par exemple une saisie incorrecte ou une erreur de base de données). Cela permet d'éviter le crash de l'application et d'afficher un message d'erreur compréhensible à l'utilisateur.

Quelle est la différence entre `setError()` et `Toast` ?

Réponse :

`setError()` affiche un message d'erreur directement sur un champ de saisie et indique clairement où se trouve l'erreur.

`Toast` affiche un message temporaire à l'écran pour informer l'utilisateur d'un événement général, comme une inscription réussie ou une erreur globale.

Pourquoi désactiver le bouton avant validation ?

Réponse :

Désactiver le bouton empêche l'utilisateur d'envoyer un formulaire invalide. Cela améliore l'expérience utilisateur, réduit les erreurs de saisie et évite des traitements inutiles ou des plantages de l'application.

Que se passe-t-il si on supprime la gestion des exceptions ?

Réponse :

Sans gestion des exceptions, l'application peut planter en cas d'erreur (saisie incorrecte, valeur non numérique, problème de base de données). Cela rend l'application instable et inutilisable pour l'utilisateur.

