

## React Native – SQLite, HTTP, Secure Store, useReducer

Vous devez refactoriser votre application de l'exercice 6. Vous devez obtenir les magasins par une requête à L'API suivant : <https://my-json-server.typicode.com/frederic-s-f/252-mock-data/stores>. Les magasins obtenus seront sauvegardés dans une base de données SQLite. Par la suite, l'application pourra fonctionner sans accès au réseau internet. Comme la librairie AsyncStorage n'est pas sécurisée et donc pas adaptée pour sauvegarder les informations de l'utilisateur (mot de passe), vous devez changer votre code de façon à utiliser le module expo Secure Store. Vous devez aussi utiliser useReducer ou Redux pour la gestion de la présentation de la page de Login et le magasin sélectionné.

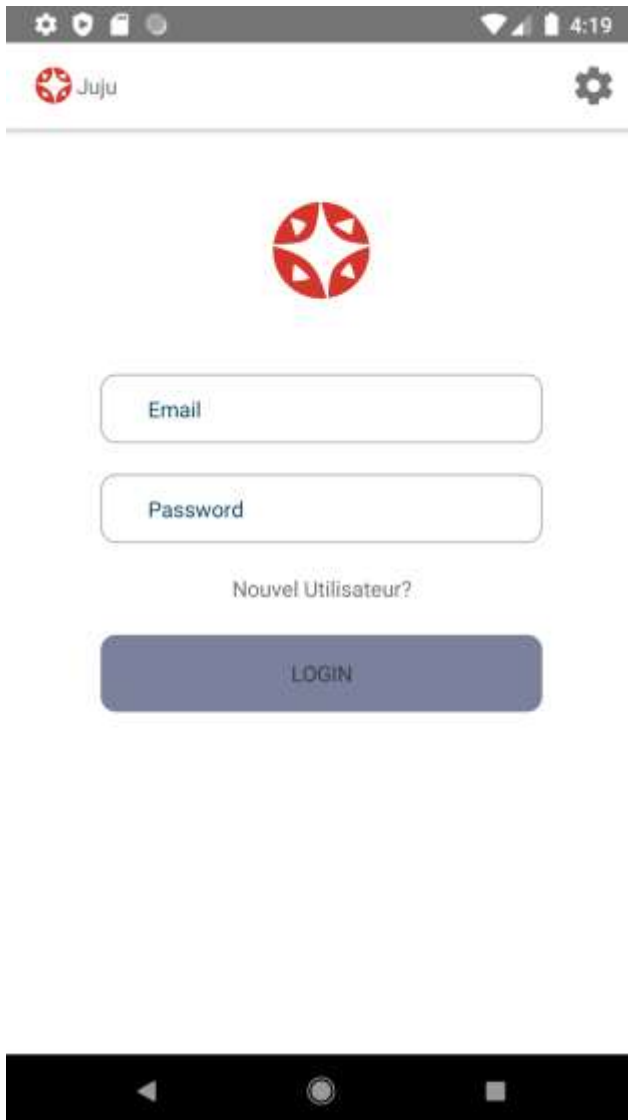


Figure 1

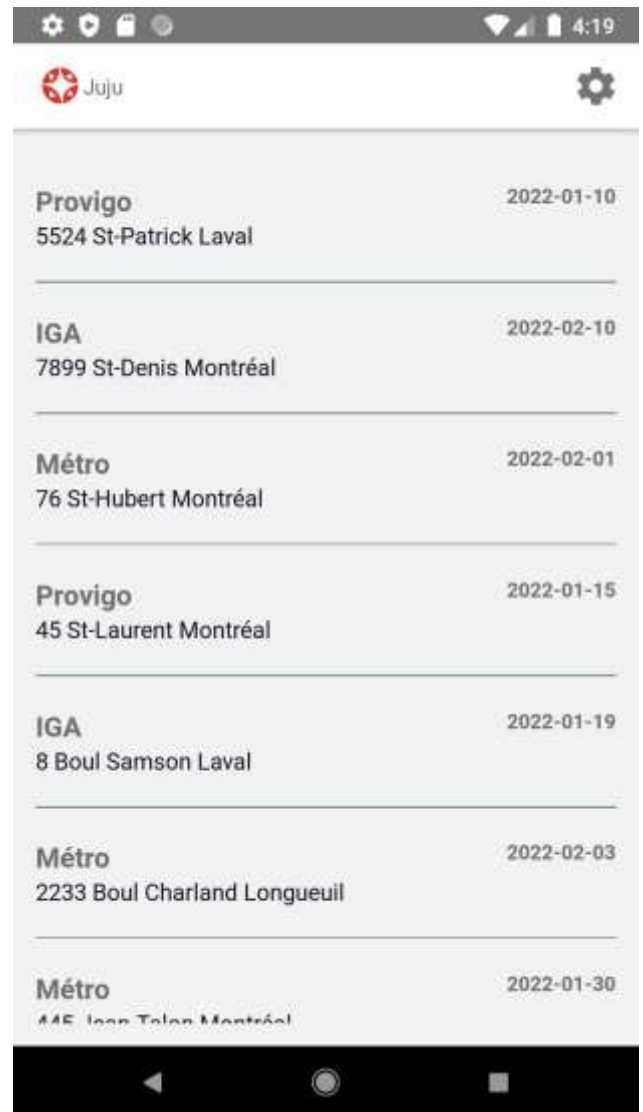


Figure 2

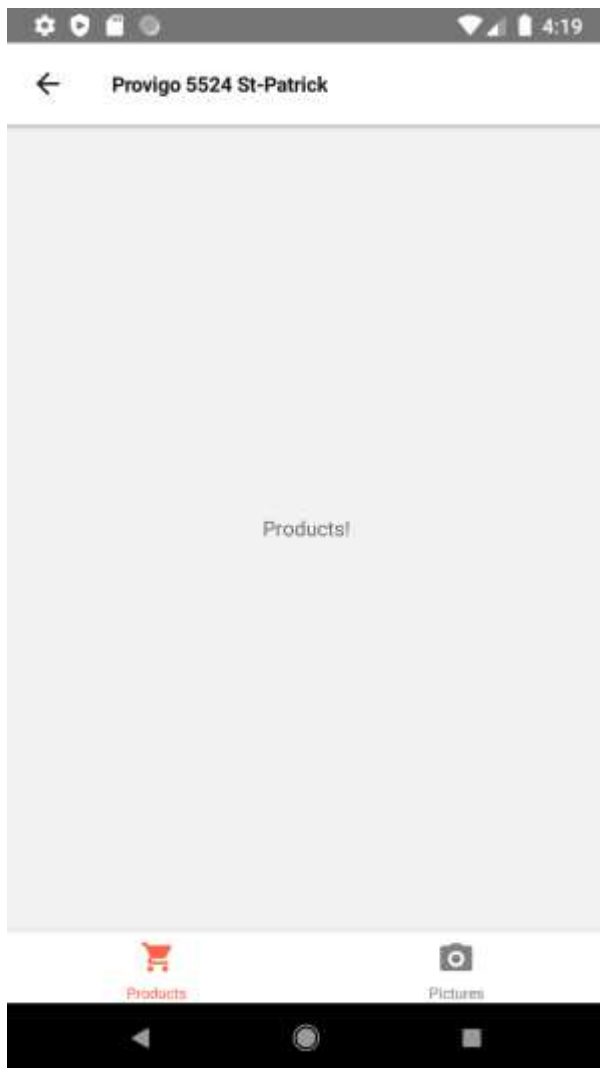


Figure 3

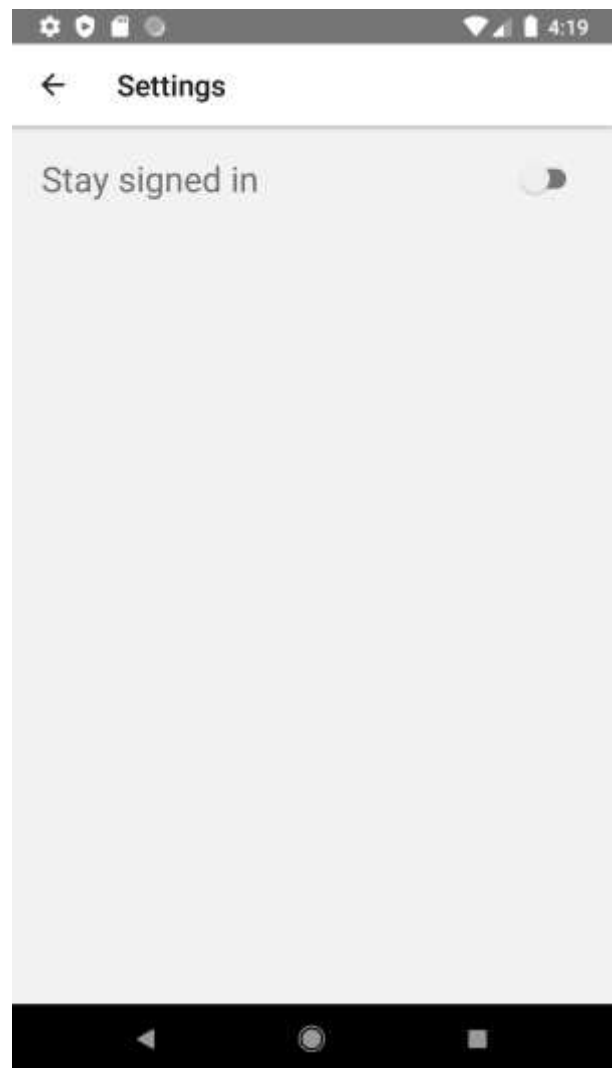
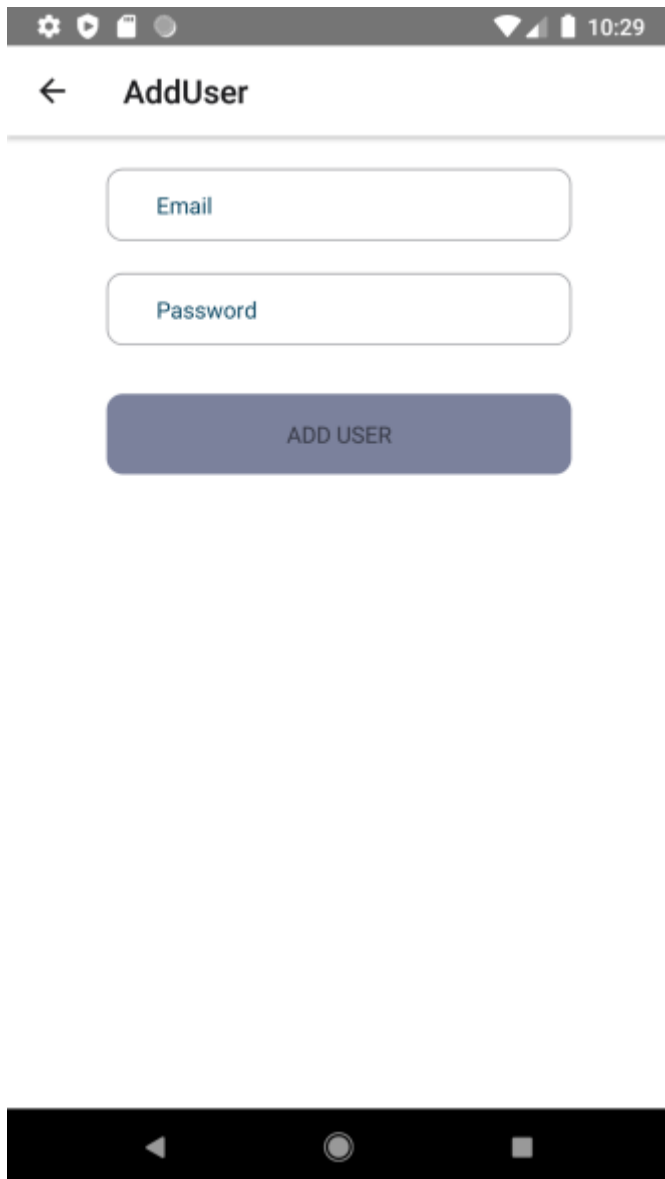


Figure 4



The image shows a screenshot of an Android application interface. At the top, there is a status bar with icons for settings, a shield, a document, and a circle, along with the time 10:29. Below the status bar is a header bar with a back arrow icon and the text "AddUser". A horizontal line separates the header from the main content area. The main content area contains two input fields: the first is labeled "Email" and the second is labeled "Password". Below these fields is a blue button with the text "ADD USER". At the bottom of the screen is a black navigation bar with three icons: a back arrow, a circle, and a square.

*Figure 5*