A SIGNER

DOSSIER D'ARCHITECTURE TECHNIQUE

Application mobile d'apprentissage de la langue des signes française (LSF)

TRIPOT Melanie, VALLIN Kévin et LAURENCINE Corentin

SOMMAIRE

Contexte

A signer est une application mobile ayant pour but l'apprentissage de la Langue des signes française (LSF) , elle s'adresse au personne entendante souhaitant apprendre la LSF.
Elle sera donc utilisable sur des appareils mobiles tels que des smartphones ou des tablettes.

Elle se présentera comme un application d'éducation linguistique semblable au fonctionnement de "Duolingo" sans la partie compréhension orale
dans l'idée de l'application "High Five" mais avec des vidéos de vraie personne et non pas des animation 3D.

Besoins principaux

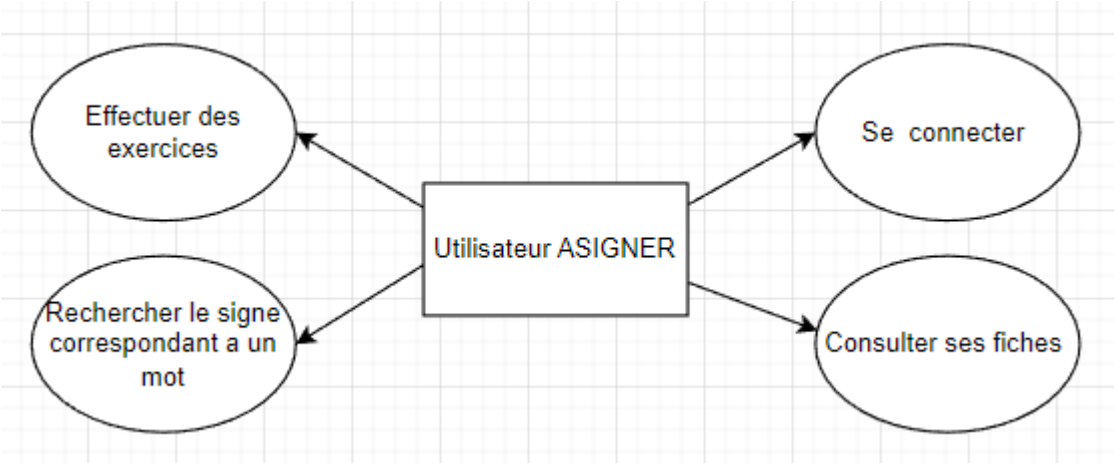
- Montrer à l'utilisateur des vidéo des Signe pour les apprendre
- Enseigner la structure des phrases en LSF
- Enseigner à signer les mots et les phrases
- Enseigner l'importance de l'expression faciale pendant les signe

L'application se découpera en plusieurs fonctionnalités :

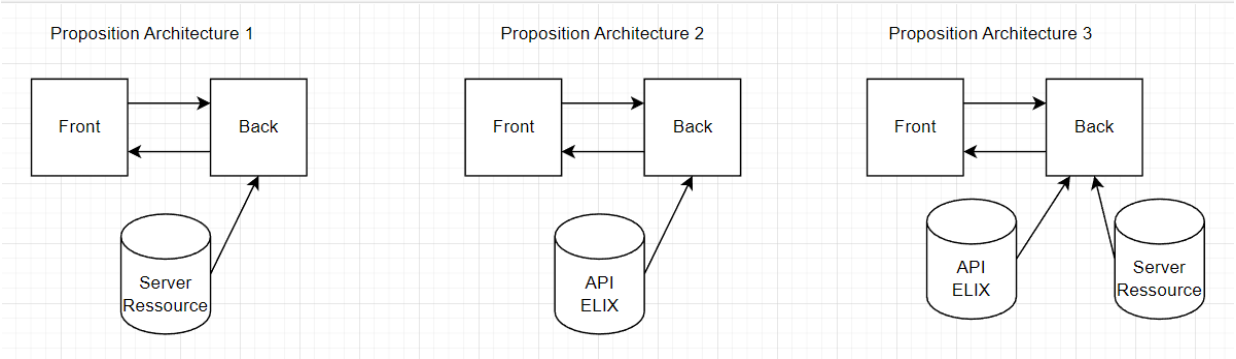
- parcours : ici l'utilisateur aura un parcour de formation à suivre sous la forme d'exercice, quand un utilisateur termine un niveau il débloque le suivant et ainsi de suite, jusqu'à la fin du parcours
- fiches : l'utilisateur pourra retrouver des signes qu'il a déjà rencontrés lors de ses exercices , le tout sera trié par thème (Emotion / Politesse / Action …), ainsi que les différentes règles de syntaxe de la LSF
- dictionnaire / signaire : permet à l'utilisateur de retrouver un/des signes associé(s) à un mot
- compte : pour pouvoir enregistrer son parcours, ainsi que les différentes activités qu'il a déjà réalisées ou non. Donner la possibilité d'accéder à son "parcours" sur n'importe quel appareil (authentification avec google ?)
- paramètres : pour pouvoir accéder à la modification de mots de passe,(potentiellement activé le F2P), changement de langue application, notifications, signaler un problème, déconnexion, A propos (Webview ?)

Fonctionnalités souhaitées

- Authentification
    - se connecter
    - mot de passe oublié
    - créer un compte
    - se déconnecter
    - Authentification via google ?
- Parcours
    - lister les Thèmes
        - lister exercices
            - montrer les vidéo / signes + consigne + proposer les réponse + valider ou non la réponse
            - exercice fait / non fait : indiquer si l'utilisateur a déjà terminer cet exercice
- Fiches
    - Lister les Thèmes
    - ouvrir les fiches
    - afficher les signes en plein écran
- Dictionnaire / Signaire
    - Rechercher un mot
    - Afficher les signes associés au mot
    - Afficher les signes en plein écran
- Paramètres
    - Paramètres légaux généraux
    - Choix du thème (défaut, dark, light)
    - Choix de la langue de l'application ?
    - Modification de la taille des polices
    - Mode accessible ?
    - Centre d'aide

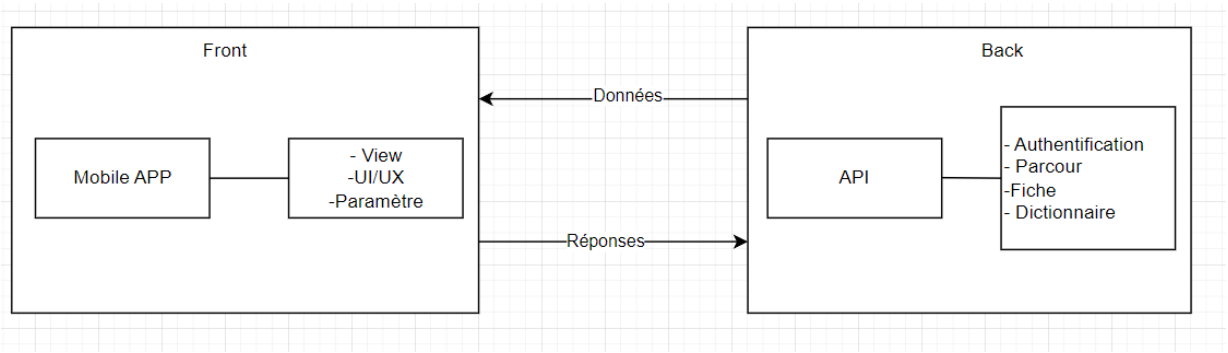

Effectuer des exercices
Se connecter
Utilisateur ASIGNER
Rechercher le signe correspondant a un mot
Consulter ses fiches

Architecture applicative / technique :

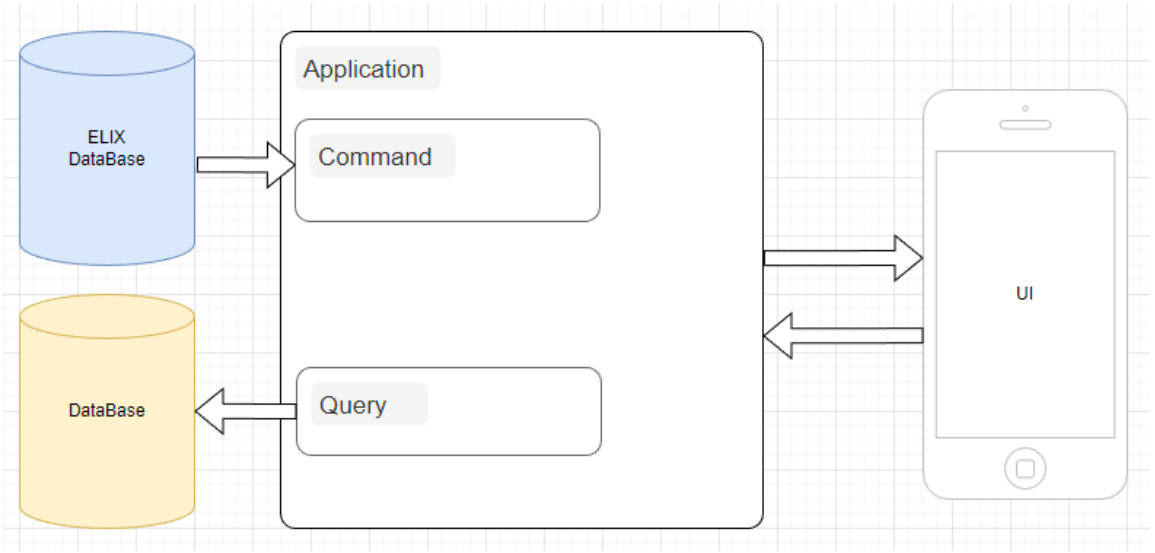


Voici trois propositions d'architecture pour le projet A Signer :

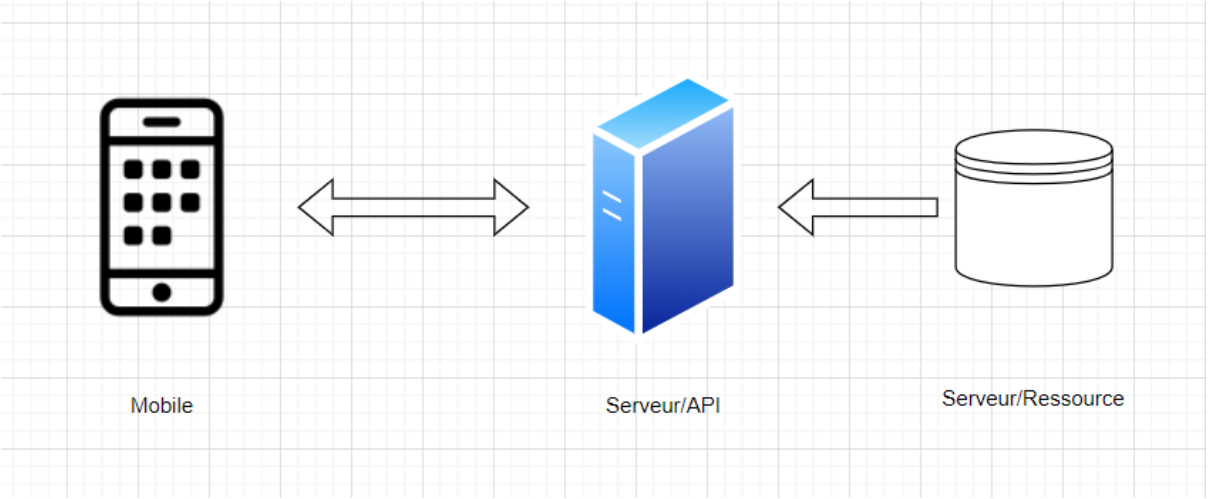
- La proposition 1 serait mise en place dans l'éventualité où un partenariat avec Le Dico Elix ne serait pas possible.
- La proposition 2 serait possible si nous parvenons à obtenir un partenariat avec Le Dico Elix
- La proposition 3 serait possible si nous parvenons à obtenir un partenariat avec Le Dico Elix mais où leurs ressources ne serait pas suffisante pour le fonctionnement complet du projet A signer



CQRS Pattern



Architecture physique / infrastructure



Environnements

Deux proposition de technologies : Flutter & React.native

Flutter Vs React Native: Quick Comparison

	Flutter	React Native
Initial release	May 2017	March 2015
Backed by	Google	Facebook
Programming language	Dart	JavaScript
Platform support	Android, iOS	Android, iOS, Web apps
Application performance	Fairly robust, 60 fps	Close to native
IDE support	Android Studio, Visual Studio Code, IntelliJ IDEA	Range of IDE's and tolls with JS support
Native appearance	Better. Has access to device core functionalities	Lower. Dependency on third party apps
Hot reloading	Yes	Yes
GUI	Use proprietary widgets and deliver UI	Use native UI controller
Time to market	Faster	Slower than Flutter
Code reusability	50-90%	90%
Testing	Mobile device or emulator	Mobile device or emulator
Community & support	Limited, Fast growing	Extensive

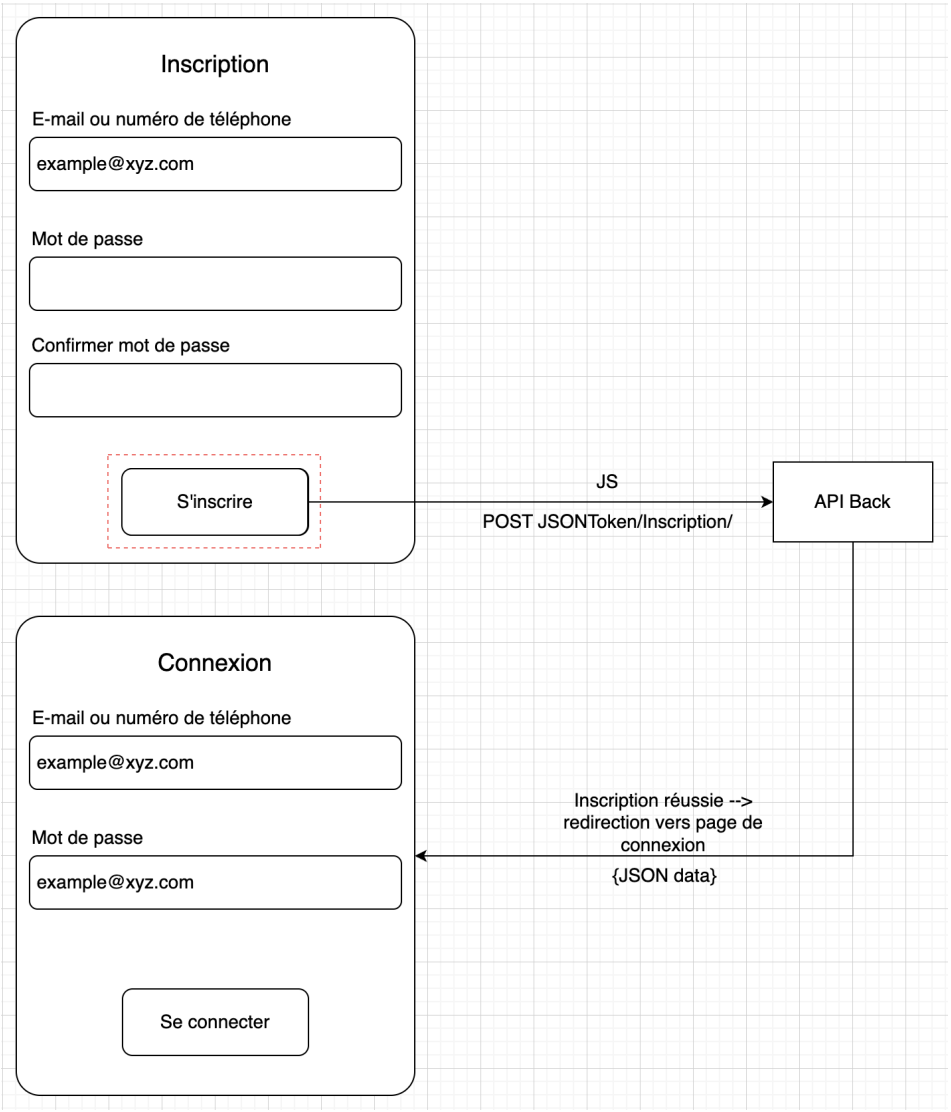
Avantage à l'utilisation de React NAtive :

- l'équipe connais déjà javascript,
- possibilité de passer sur une Web app

Détail fonctionnalités principales souhaitées

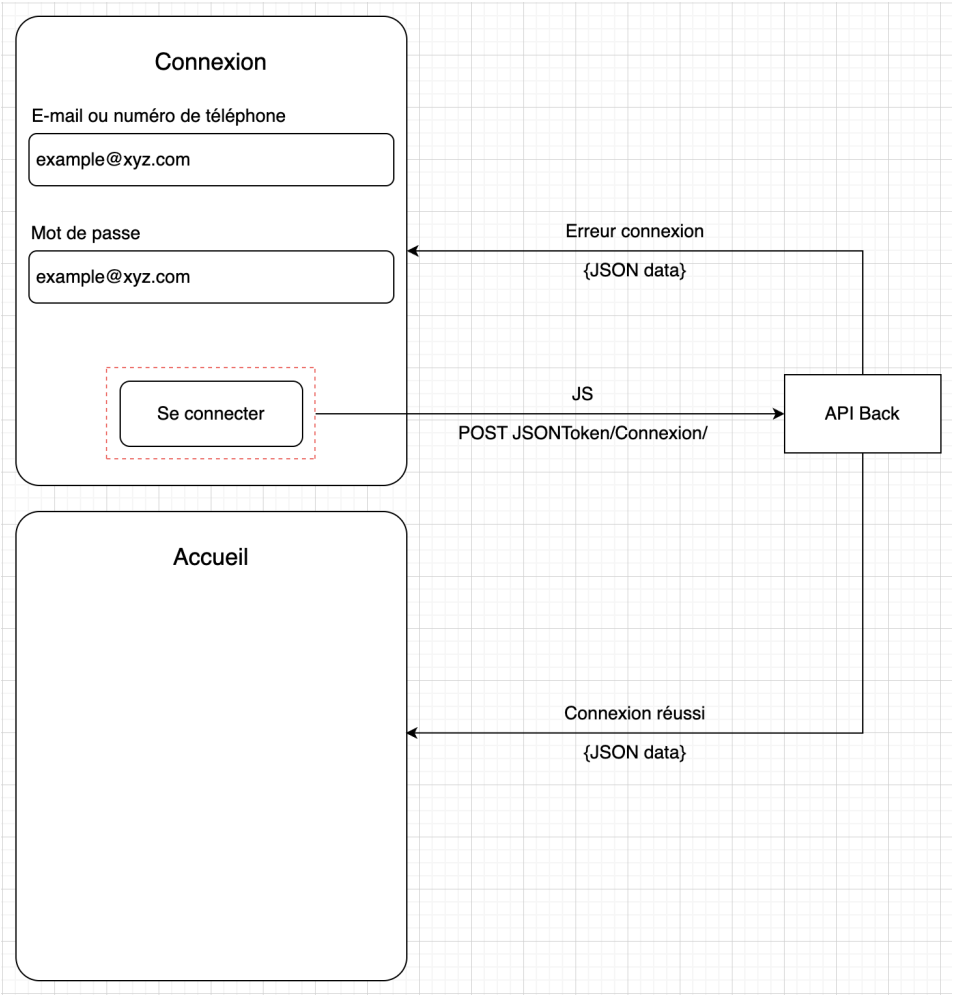
Détail Authentification

Modèle d'inscription utilisateur :



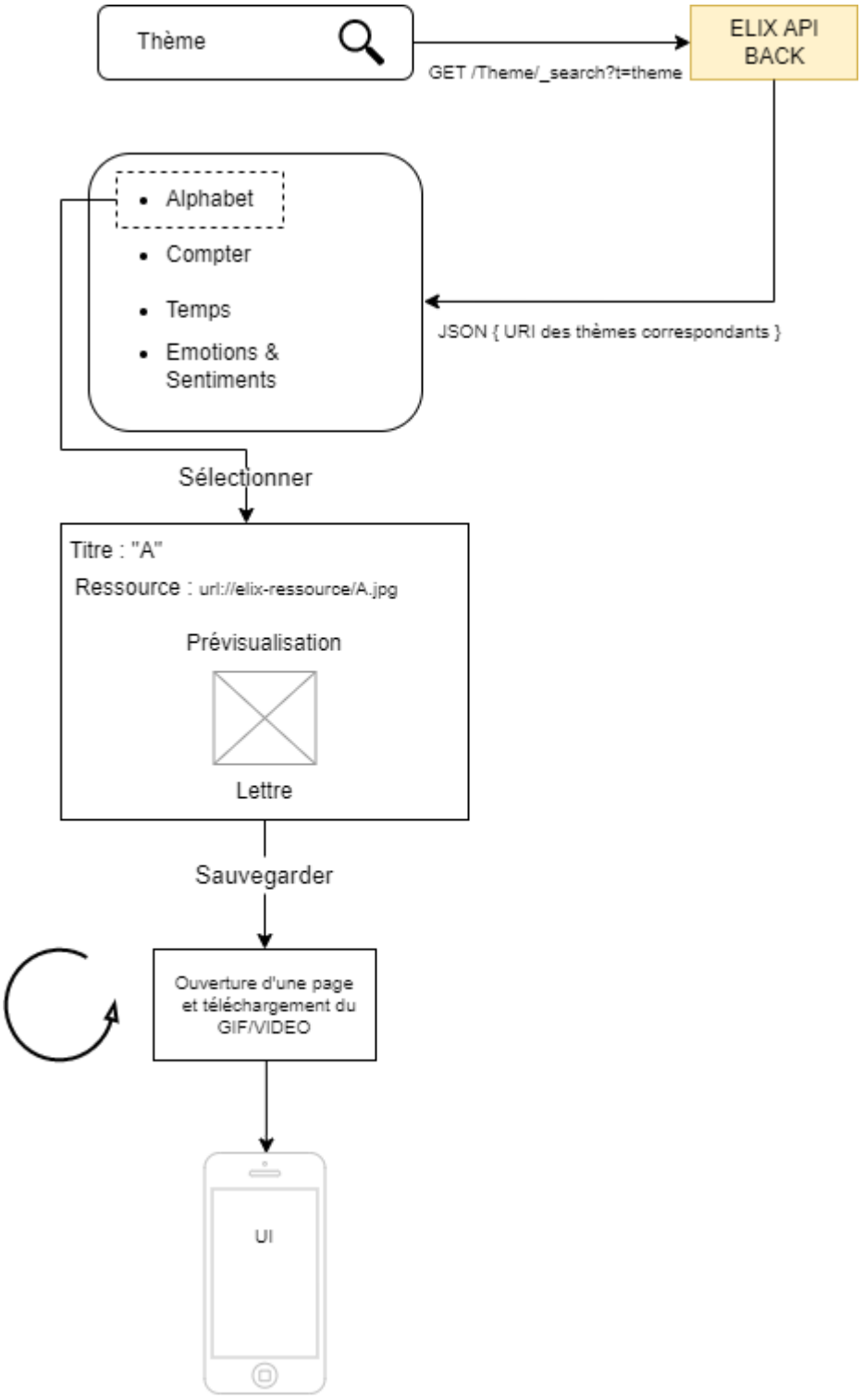
Lors de l'inscription, le login et le mot de passe de l'utilisateur sont stockés en base de données (BDD) et le mot de passe est hashé pour ne pas qu'il apparaisse en clair si on regarde les données de la table correspondante.

Modèle de connexion utilisateur :



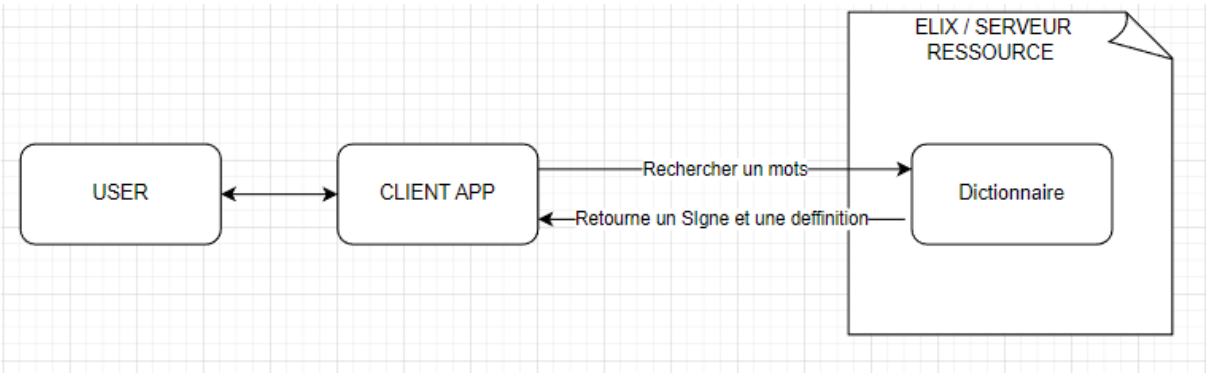
Lors de la connexion, le mot de passe est comparé au niveau de la communication avec l'API pour vérifier sa justesse.
Si il est incorrect, un message d'erreur s'affiche sur la page de connexion.
Si il est correct, l'application redirige l'utilisateur vers la page d'accueil de l'application.

Details partie Fiche



Details partie dictionnaire/ Signaire

Dans les deux possibilité d'utilisation de l'API ELIX ou non la visualisation de la partie Dictionnaire devrait avoir la forme suivante



Opérationnel :

Procédures d'installation, backup, …