# PROJET DE LICENCE

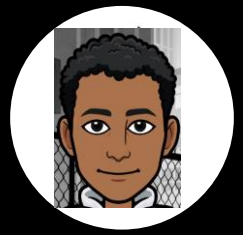
## DÉVELOPPEMENT D'UNE APPLICATION ANDROID SERVEUR PERMETTANT LA CRÉATION DE PARCOURS ETUDIANTS L1 ET L2

## NOTRE ÉQUIPE

*Florian SPIRE*

*Sofian MOUMEN*

*Dimitri MORELLE*

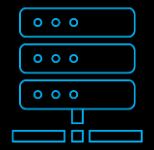


*Ossama ASHRAF*

*Jeremy HIRTH DAUMAS*

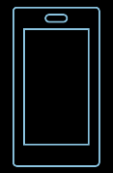
# BILAN DES FONCTIONNALITÉS

Application Android permettant aux étudiants de se créer un parcours de L1 et de L2. Conception d'un serveur permettant une communication application-serveur fonctionnelle.



Les fonctionnalités serveur comprennent :

- Enregistrement des parcours
- Envoi des UE (+ descriptions), prérequis et parcours prédéfinis en temps réel en fonction des demandes du client
- Enregistrement des étudiants
- Gestion de la connexion des étudiants (y compris connexions persistantes)
- Gestion des mots de passe oubliés

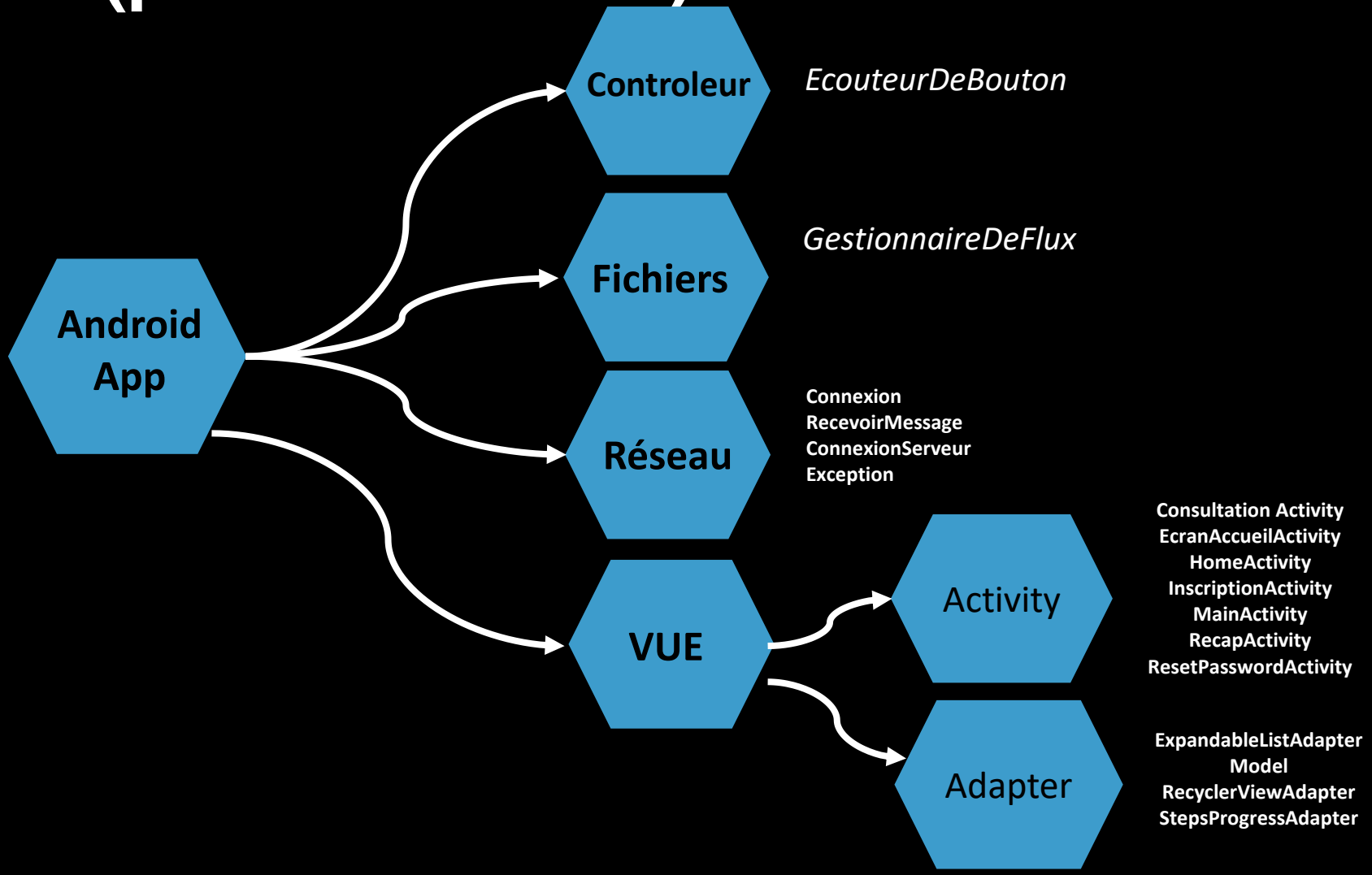


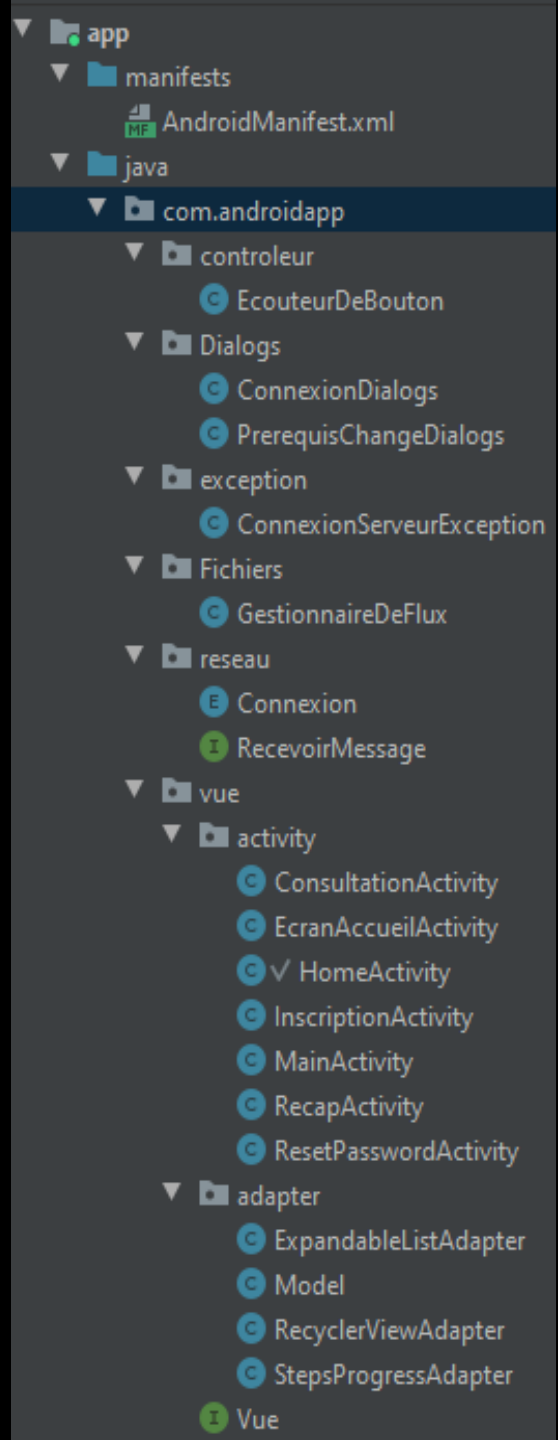
Les fonctionnalités AndroidApp comprennent :

- Consultations des parcours des autres étudiants
- Conceptions de parcours (personnalisés ou prédéfinis)
- Inscriptions et connexions d'étudiants
- Partages de parcours
- Possibilité de récupération de mot de passe grâce à la combinaison date de naissance / mot de passe
- Affichage de la description des UE par appui long sur celles-ci

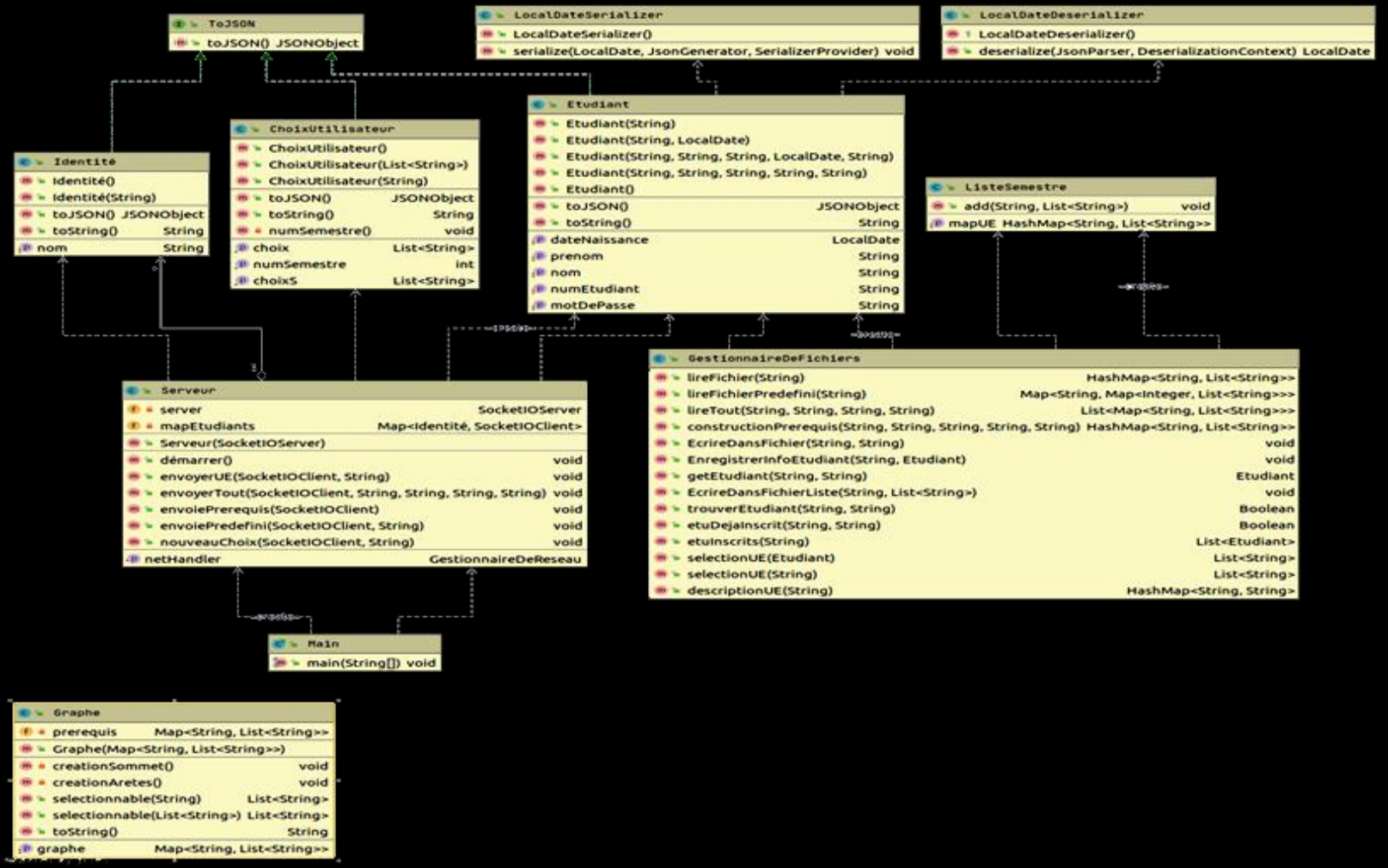
Interface graphique la plus organisée et intuitive possible, regroupant les UE par disciplines, coloration des UE sélectionnées etc.

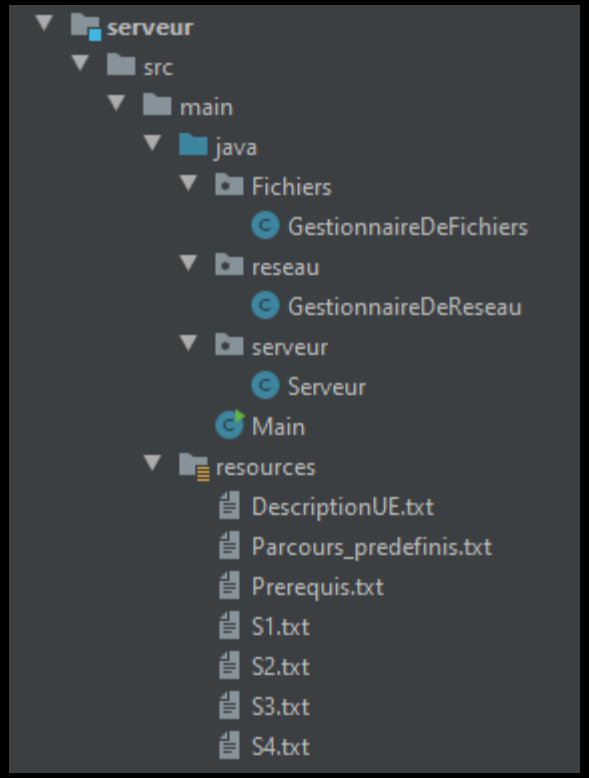
# CONCEPTION ET ORGANISATION DU CODE (partie Android)



# CONCEPTION ET ORGANISATION DU CODE (partie Serveur)



# TESTS UNITAIRES ET IHM


## IHM

- Réalisées pour chaque itération
- Revues au fur et à mesure de l'evolution du projet
- Ressemblantes au résultat final
- Permettent de se fixer des objectifs à réaliser pour chaque livraison

## TESTS UNITAIRES

- Tests graphiques avec Espresso (on simule le comportement d'un utilisateur lambda)
- Utilisations de mocks
- Tests classiques de chaque classes côté serveur et classes communes client-serveur

## POINTS FORTS VS POINTS FAIBLES

- Tous réalisés
- N'échouent pas

- Possibilité de rendre plus complets les tests
- Dépendance des tests au serveur

# POINTS FORTS ET POINTS FAIBLES

## POINTS FORTS

- **Gestion évènementielle client-serveur**
  - *Envoi des UE*
  - *Envoi des prérequis*
  - *Envoi des parcours pour consultation*
  - *Envoi de mot de passe en cas d'oubli*
- **Bonne modularité du serveur**
  - *Ajout d'événements simples*
  - *Bonne organisation simple à comprendre*
- **Bien organisé par types**
  - *Par activités/adapter*
  - *Séparation classes communes avec le serveur et locales*
  - *Objets adaptés : étudiants, map, localDate, …*

## POINTS FAIBLES

- **Densité des activités Android :**
  - *En particulier mainActivity et ecranAccueilActivity*
  - *Un nouveau développeur pourrait mettre du temps à s'approprier le code*
- **Connexion serveur obligatoire pour toutes les opérations**
- **Tests unitaires dépendants du serveur**

## DIFFICULTÉS

La répartition du travail dans le groupe

La gestion des modifications du sujet.

Le nombre de tests et de fonctionnalités à livrer

## LIVRAISONS

Dans les temps

Reports réguliers de fonctionnalités aux itérations suivantes

Mais toutes ont été implémentées dans la livraison finale !

# GESTION DU PROJET

*Démonstration…*