Soutenance de projet

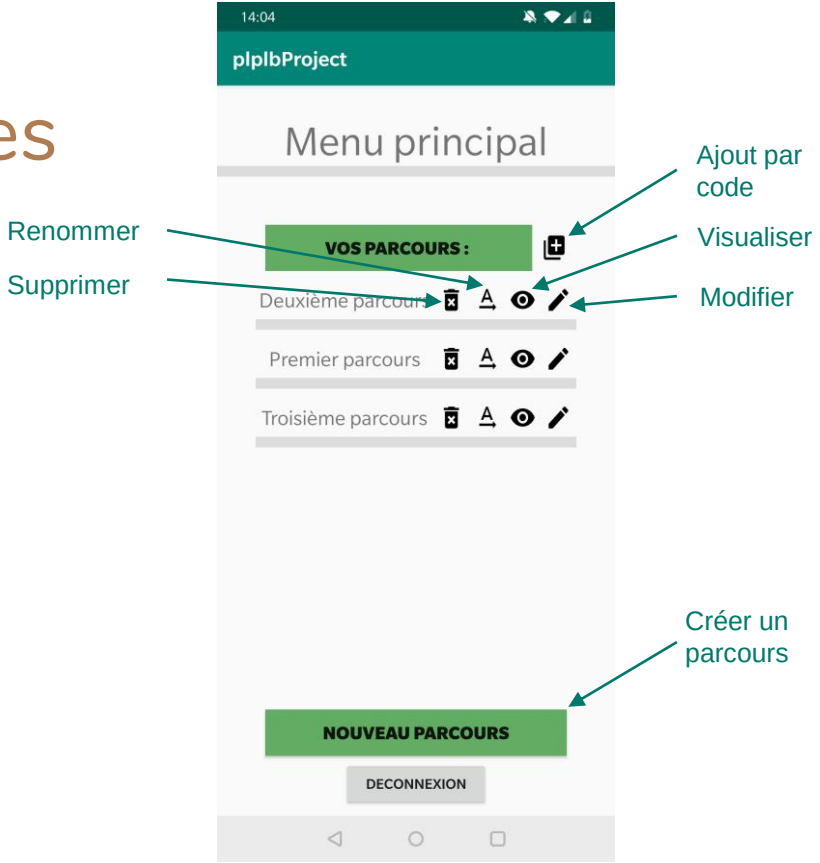
Goulot Thomas, Kugler Romain, Jerome Maxime, Martin D’Escrienne Yann, Tognetti Yohann

Fonctionnalitées réalisées

- Application android fonctionnelle:
  - **Page de connexion**

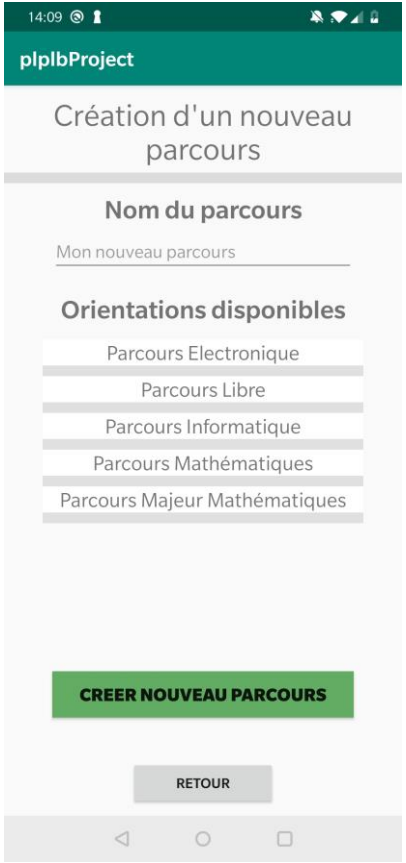
Fonctionnalitées réalisées

- Application android fonctionnelle:
  - Page de connexion
  - **Menu Principal**



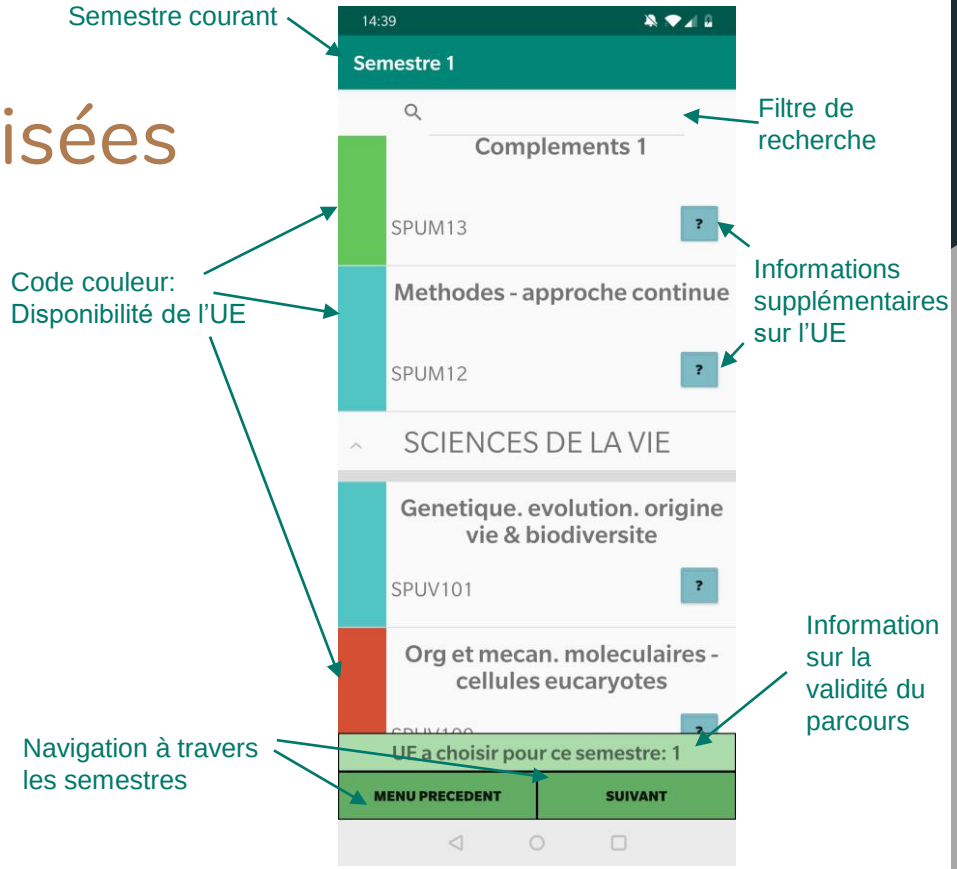
Fonctionnalitées réalisées

- Application android fonctionnelle:
  - Page de connexion
  - Menu Principal
  - **Créer un nouveau parcours**



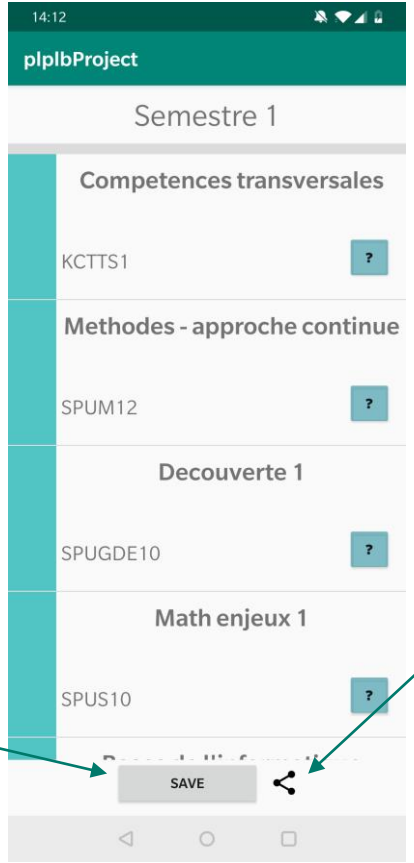
Fonctionnalitées réalisées

- Application android fonctionnelle:
  - Page de connexion
  - Menu Principal
  - Créer un nouveau parcours
  - **Construire un parcours (4 semestres)**



Fonctionnalitées réalisées

- Application android fonctionnelle:
  - Page de connexion
  - Menu Principal
  - Créer un nouveau parcours
  - Construire un parcours (4 semestres)
  - **Aperçu du parcours (Accessible depuis le menu principal et à la fin de la création de parcours)**



Sauvegarder le parcours (disponible que si le parcours vient d'être créé)

Partage du parcours

Fonctionnalitées réalisées

- Application android fonctionnelle:
  - Page de connexion
  - Menu Principal
  - Créer un nouveau parcours
  - Construire un parcours (4 semestres)
  - Aperçu du parcours (Accessible depuis le menu principal et à la fin de la création de parcours)
  - **Partage de parcours**

Partage du parcours

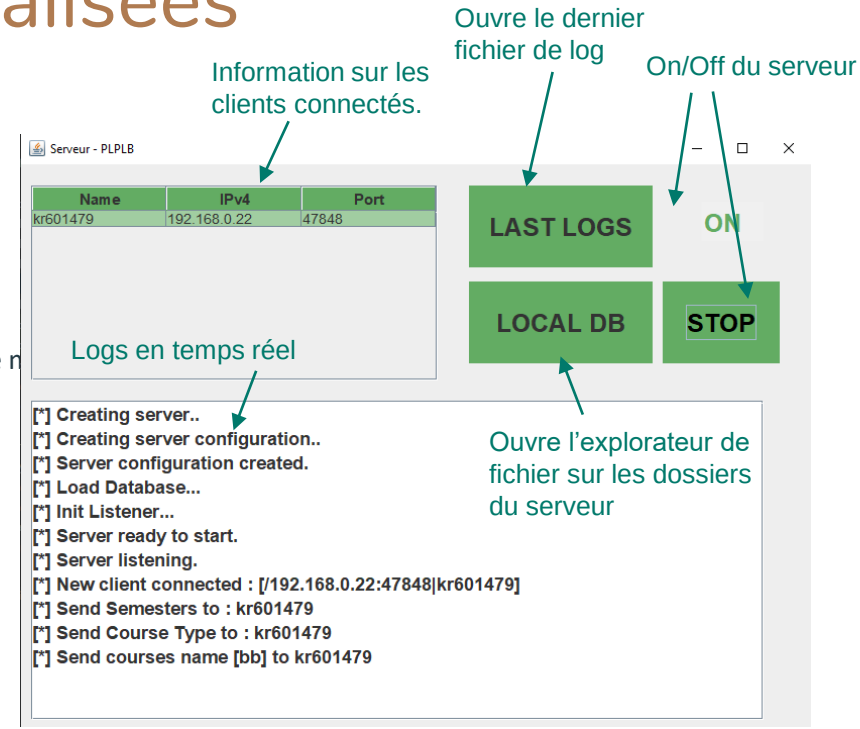
Partager le code en utilisant une messagerie.

Partager le code en utilisant diverses applications (discord, messenger).

Fonctionnalitées réalisées

- Application android fonctionnelle:
  - Page de connexion
  - Menu Principal
  - Créer un nouveau parcours
  - Construire un parcours (4 semestres)
  - Aperçu du parcours (Accessible depuis le n
    et à la fin de la création de parcours)
  - Partage de parcours

- Côté serveur:
  - Gestion de cas d'erreur
  - Gestion des bases de données et de logs
  - Interface graphique
  - Fichier de configuration



Choix de conception, Organisation du code

- Organisation en packages:
  - Android (client) : Code de l'application elle-même divisée en package (vues, contrôleurs, Réseau …)
  - Javastd (serveur) : Code du serveur (package) sauvegardes et codes partagé (modèles, listeners).
- Utilisation de design patterns:
  - Adapters (évident)
  - Singleton pour la connexion

Choix de conception, Organisation du code

- Serveur le plus optimal et mutable possible :
  - Envoie du minimum d'événement possible au serveur.
  - Fichier config pour changer rapidement les paramètres.
- Stockage des données :
  - Fichier .csv pour les ues afin de changer les champs rapidement.
  - Dossiers séparés pour les sauvegardes, les parcours partagés…

Organisation des tests

- Côté client:
  - Tests indépendants du serveur: le serveur n'est **jamais** sollicité.
  - Utilisation de Mockito pour simuler le fonctionnement de l'application:
    - On simule des retours de fonctions. (*mock.then return)*
    - On simule des interactions avec l'IHM (*perform()*)

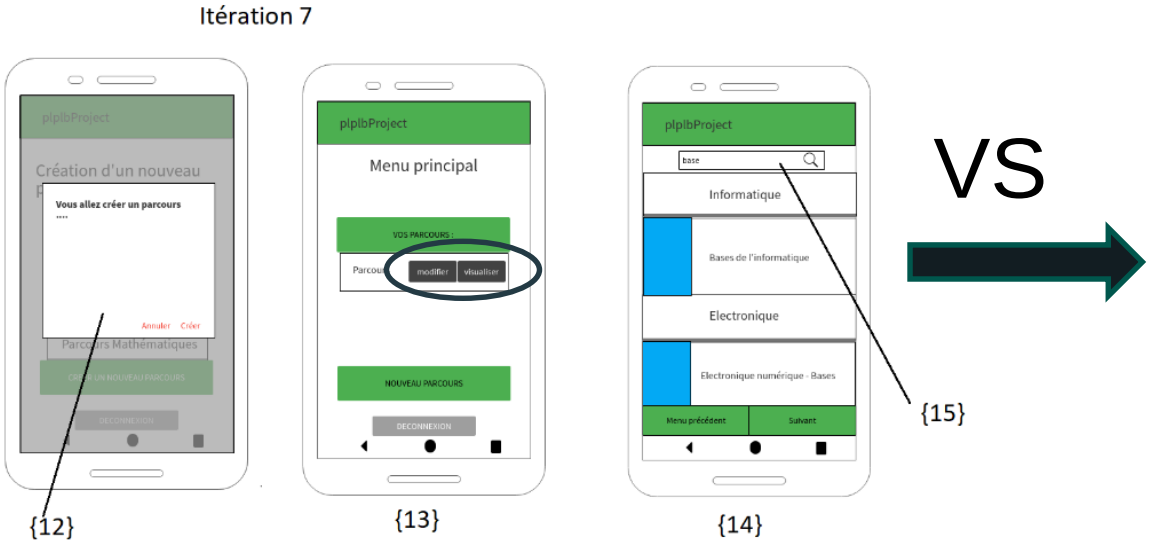
Evolution des IHMs

- Evolution de l'IHM conforme à notre maquette …

VS

Evolution des IHMs

- … avec certaines évolutions



Points forts et points faibles de l'implémentation.

- Point forts
  - Stabilité, prévention de bugs
  - Tests java complets
  - Modularité et dissociation
  - Compréhensibilité et lisibilité
- Point faible
  - Ergonomie
  - Beaucoup de refactoring lié à un manque d'anticipation des itérations.
  - Test graphiques pouvant être plus poussés

Gestion du projet

- Mauvaise prévision sur les itérations.
- Retard rattrapé sur les tests.
- Certaines évolutions demandées ont été anticipées.
- Participation relativement équivalente.
- Spécialisation dans les implémentations.

Démo

En direct…