

Réalisation d'une application Android :

VDE (Vie des ELI)

Contraintes pédagogiques

- Réutiliser certains des éléments mis en œuvre durant les travaux pratiques ;
- Concrétiser l'apprentissage avec l'utilisation de possibilités Android non abordées durant les séances ;
- Utilisation de connaissances réseaux ;

Table des matières

VDE (Vie des ELI).....	1
Contraintes pédagogiques	1
But de l'application.....	2
Fonctionnalités.....	2
Fonctionnalités vitales.....	2
Interfaces utilisateurs (activities) :.....	2
Stockage et accès aux données :.....	2
Soumettre une anecdote :.....	2
Afficher les anecdotes :.....	2
Fonctionnalités supplémentaires.....	3
Interfaces utilisateurs (activities) :.....	3
Stockage et accès aux données :.....	3
Soumettre une anecdote :.....	3
Afficher les anecdotes :.....	3
Autre :.....	3
Choix techniques particuliers.....	4
Serveur.....	4
Multi-threadage des requêtes serveur.....	4
Système de préférences.....	4
Apprentissage.....	4
Vue en tronc commun (toutes des compétences Android).....	4
Apprentissage du projet.....	4
Schéma fonctionnel de l'application.....	6
Screenshots.....	7
UML.....	8
Organisationnel.....	8
Gantt.....	8
Collaboration.....	8

But de l'application

Le but de l'application Android est de permettre à tous les étudiants ELI de poster et de consulter des anecdotes de la vie d'un étudiant ELI sur son téléphone Android.

Le niveau de SDK minimum visé est le SDK 8, soit Android Froyo (2.2).

Fonctionnalités

Parmi les fonctionnalités à implémenter, on distingue deux catégories : les fonctionnalités nécessaires au fonctionnement minimal de l'application et celles dont la non présence ne gênera pas la réalisation du dernier point.

Fonctionnalités vitales

Interfaces utilisateurs (activities) :

- Affichage d'une activité avec des boutons de navigation ;
- Activité soumettre une anecdote ;
- Activité liste les anecdotes ;

Stockage et accès aux données :

- Mise en place d'un serveur ;
- Création de la base de donnée d'anecdotes ;
- Implémentation d'un minimum de deux requêtes serveur pour l'ajout et la récupération d'anecdotes ;

Soumettre une anecdote :

- Affichage de l'activité ;
- Remplissage du formulaire pour l'ajout d'une anecdote ;
- Envoi de l'anecdote au serveur ;

Afficher les anecdotes :

- Créer une liste permettant leur visualisation
- Récupérer toutes les anecdotes ;
- Afficher les anecdotes ;

L'intégralité des fonctionnalités vitales est implémenter pour le projet. Leur raffinement à d'ailleurs aussi eu lieu grâce aux fonctionnalités supplémentaires présentes dans l'application.

Fonctionnalités supplémentaires

Ci-dessous sont présentées toutes les fonctionnalités supplémentaires envisagées.

Sont en gras celles déjà implémentées.

Interfaces utilisateurs (activities) :

- **Création et amélioration du design ;**
- **Affichage de la dernière anecdote en accueil ;**
- **Implémentation d'un menu pour le choix de l'URL du serveur ;**
- Activité d'identification utilisateur ;

Stockage et accès aux données :

- **Implémentation de requêtes spécifiques (sélection des anecdotes par page, tri par date, tri par nombre de « likes », sélection aléatoire, ajout d'un « like » à un post) ;**
- **Implémenter un objet spécifique pour les requêtes serveurs**
- **Multi-threader les requêtes serveurs ;**
- **Étoffer la représentation d'une anecdote en mémoire (ajout d'un champ « likes » à la manière Facebook...) ;**
- **Mettre en attente l'utilisateur avec une barre de défilement ;**
- Étoffer encore la représentation d'une anecdote en mémoire (ajout d'un auteur, d'une catégorie...) ;
- Création de la base de donnée utilisateurs ;
- Création d'une base de donnée par catégorie d'anecdote ;

Soumettre une anecdote :

- **Recevoir dans l'application un message de confirmation de soumission par le serveur ;**

Afficher les anecdotes :

- **Affichage par onglets des anecdotes (mieux notés (« likes »), plus récents, aléatoires) ;**
- **Paginer les affichages ;**
- **Possibilités de « liker » une anecdote une et une seule fois ;**
- Affichage par catégorie ;

Autre :

- **Mise en place d'un système de préférences pour accès aux variables préférentielles dans lors du codage de l'application ;**

Choix techniques particuliers

Serveur

Pour le serveur, il a été question d'implémenter un serveur LAMP pour mettre en commun les connaissances acquises lors du module Réseaux et Applications du cursus ELI4 avec celles de ce module.

Par ailleurs et surtout, il s'agit là du type de serveur le répandu pour ce genre d'application.

En effet, il est léger, libre, simple et réponds à toutes les attentes nécessaires à la réalisation de notre projet.

Des scripts PHP et des requêtes MySQLI ont donc été mis en place pour répondre aux besoins du projet.

L'accès aux données se faisant donc par des requêtes HTTP.

Multi-threadage des requêtes serveur

Ce multi-threadage est à la fois un moyen de protéger notre application contre les bugs serveurs et les bugs réseaux mais aussi et surtout un choix non optionnel pour les requêtes HTTP qui ne sont pas acceptés par Android en dehors d'un thread secondaire pour les versions les plus récentes des API Android.

Aussi, le choix s'est alors orientée vers la mise en place d'AsyncTask pour lesquelles la communication de données avec le fil d'exécution principal est beaucoup plus aisée notamment.

Système de préférences

Apprentissage

Voici les éléments ayant été sujets à l'utilisation et à l'acquisition de compétences spécifiques Android (en gras) ou non, ne sont pas mentionnés les éléments pour lesquels la maîtrise est nécessaire avant même de pouvoir prétendre à l'écriture d'une application :

Vue en tronc commun (toutes des compétences Android)

- Utilisation d'AsyncTask ;
- Mise en place d'une ListView ;
- Utilisation d'un menu ;
- Connaissance réseau : BufferedReader ;
- Utilisation de ProgressDialog ;
- Utilisation de différentes méthodes du cycle de vie de l'application pour une expérience utilisateur enrichie ;

Apprentissage du projet

- Mise en place de layout complexes (onglets) ;
- Mise en place d'un design ;

- **Utilisation de requêtes HTTP ;**
- **Utilisation poussée des AsyncTask ;**
- **Mise en place d'un système de préférences pour accès aux variables préférentielles ;**
- **Utilisation des TabActivity ;**
- **Utilisation de données et d'objets JSON côté Android ;**
- **Utilisation exhaustive des ressources Android : drawables, strings, colors, arrays, menus ;**
- **Codage d'une application multi-threads ;**
- **Création d'un design ;**
- Mise en place d'un serveur LAMP ;
- Approfondissement des connaissances sur phpMyAdmin ;
- Écriture de scripts PHP ;
- Apprentissage de la nouvelle API MySQLI, remplaçante de MySQL ;
- Utilisation de requêtes limitatives SQL non vues en ELI3 ;
- Utilisation de données et objets JSON côté serveur ;
- Utilisation d'un outil de collaboration : Git ;
- **Connaissance générale ;**

Schéma fonctionnel de l'application

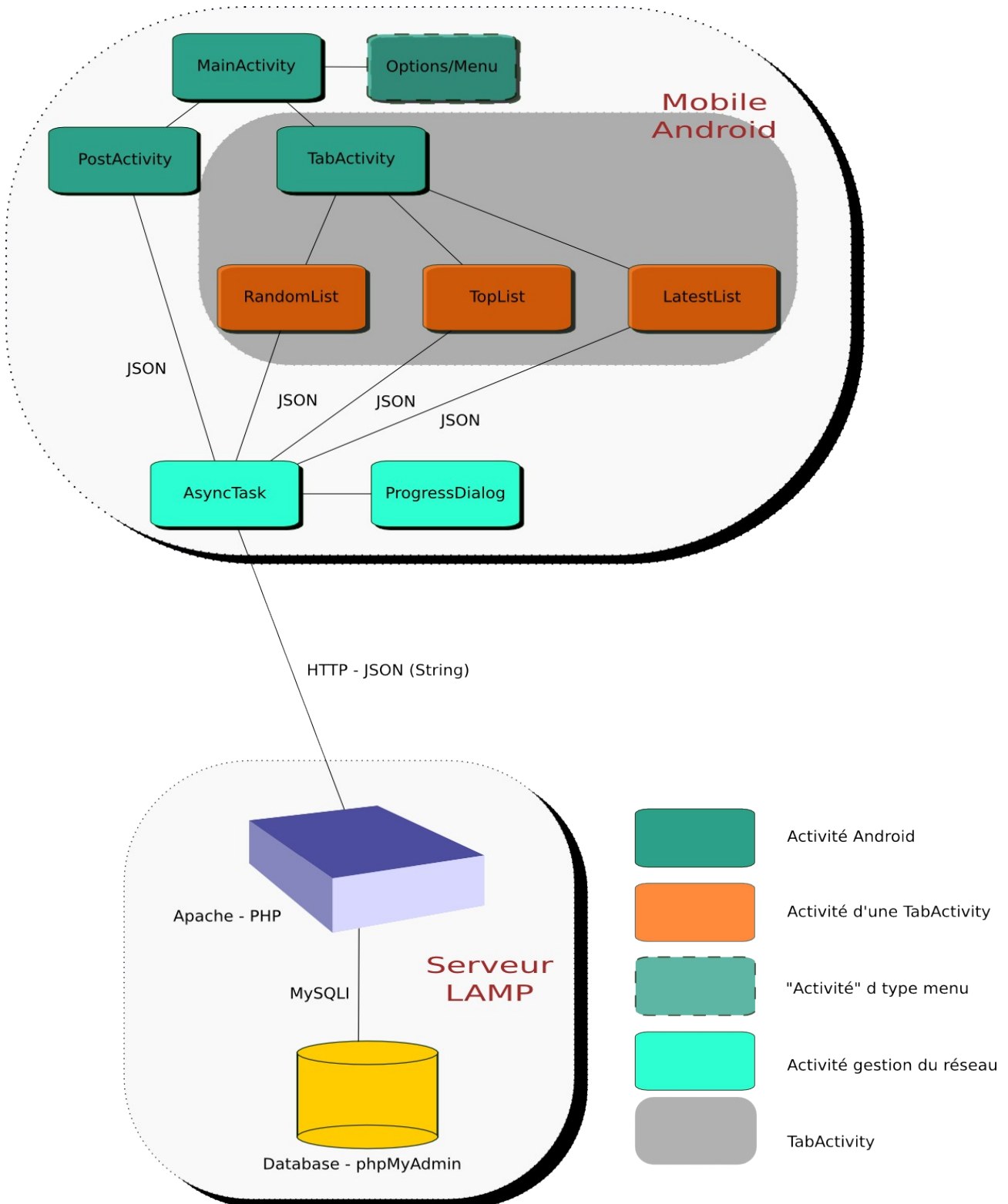


Illustration 1: schéma fonctionnel de l'application

Sur ce schéma tout n'est pas représenté mais seulement les éléments permettant d'avoir une réelle vision du projet dans sa globalité.

Screenshots

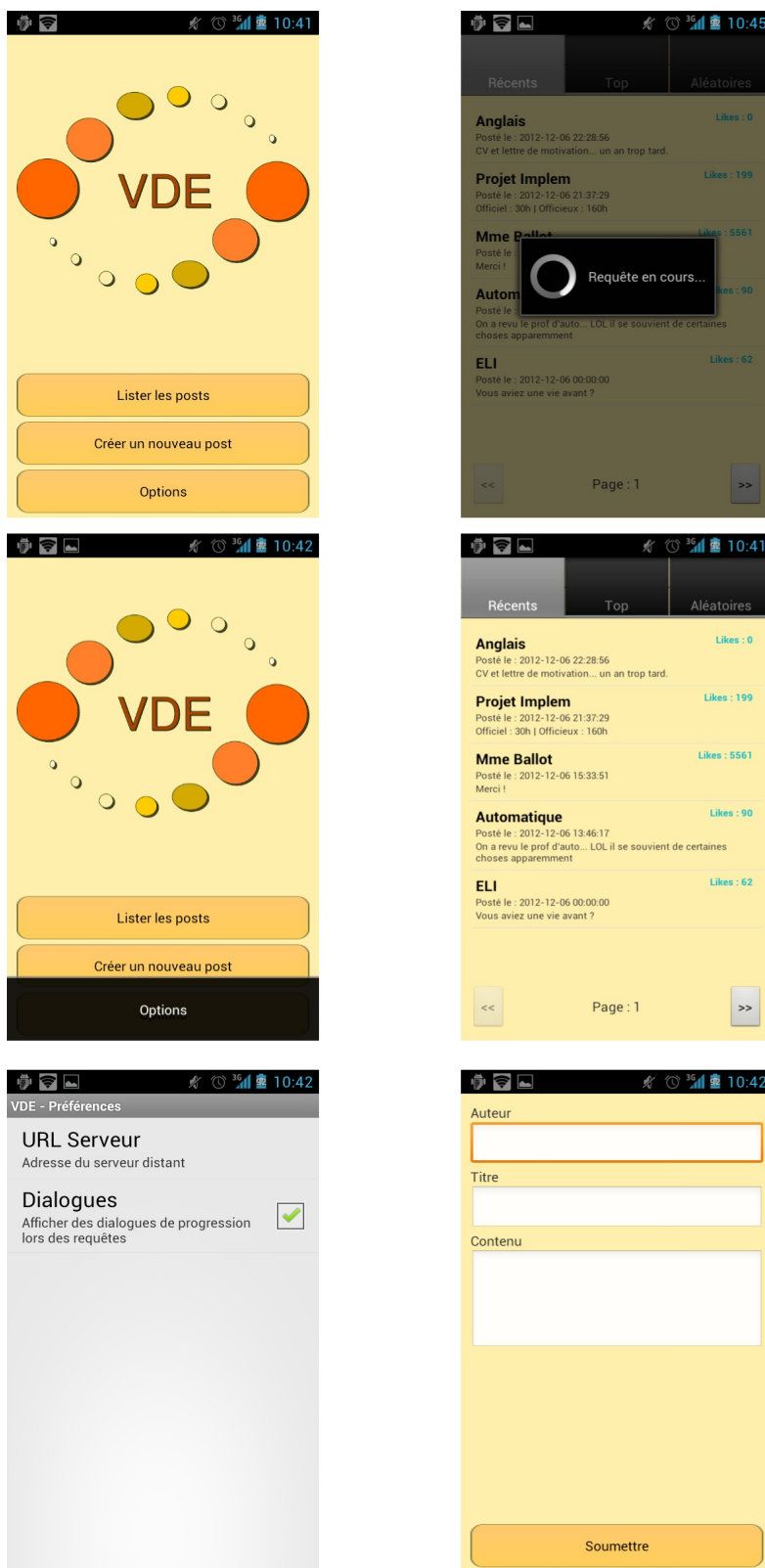


Illustration 2: Différents écrans de l'application

UML

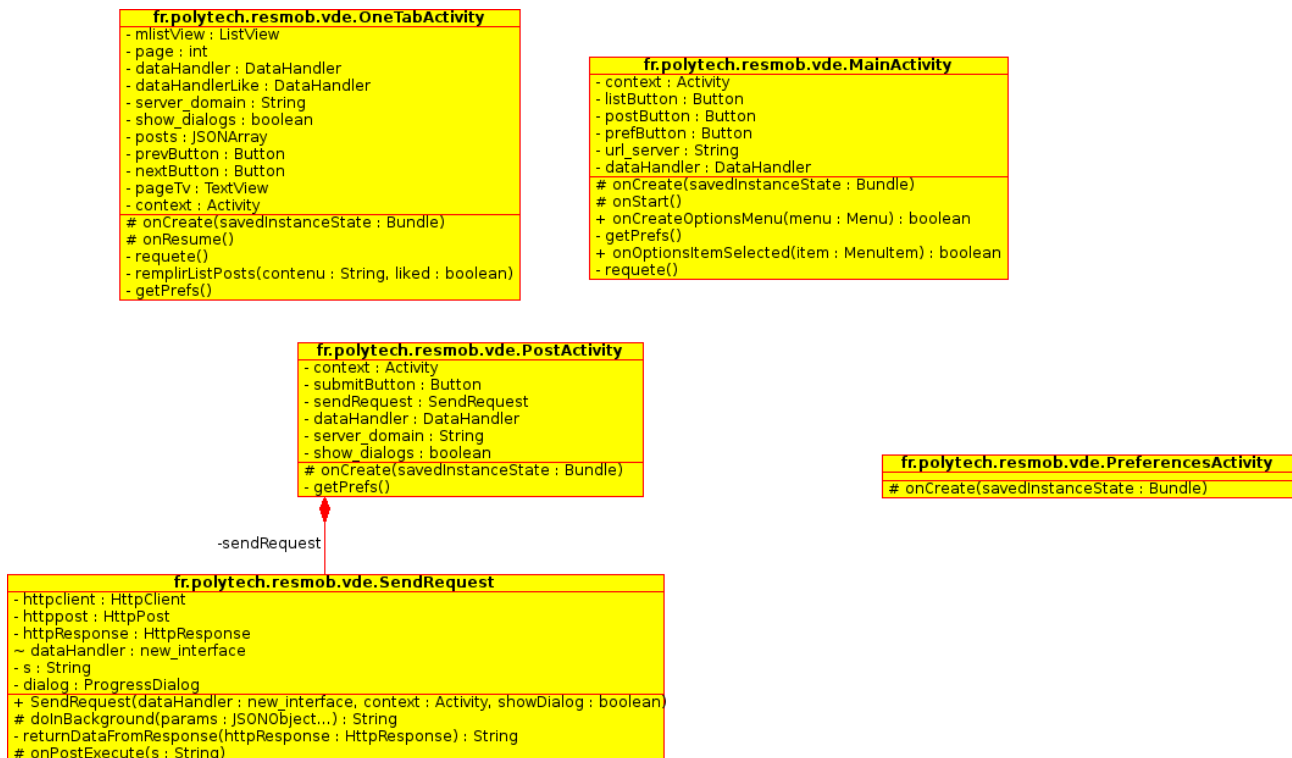


Illustration 3: Diagramme UML du projet

Organisationnel

Gantt

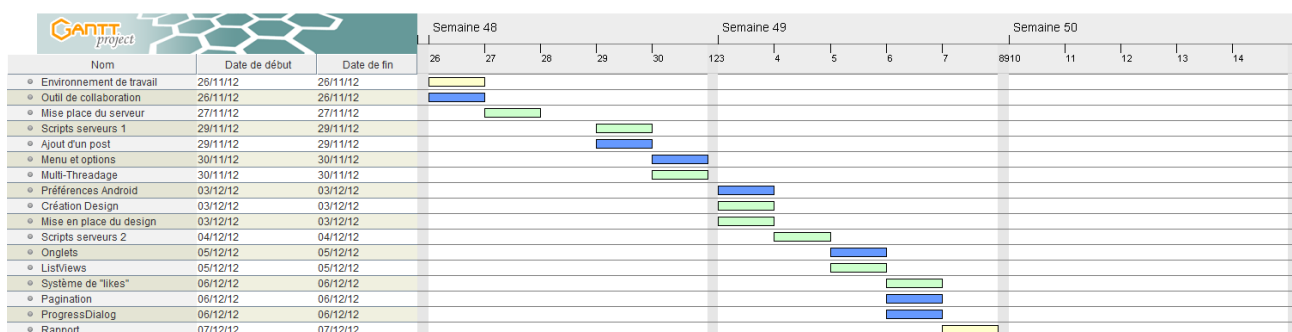


Illustration 4: Diagramme de Gantt : en bleu : Sylvain LESNE, en vert : Freddy Jaffré, en blanc : les deux.

Collaboration

Un outil de collaboration a été utilisé : Git.

Celui-ci nous a servi à versionner le code, gérer les conflits, suivre l'évolution du projet...