

AM4L: Applications Mobiles
Réalisation d'une application Android
ECAlendar

ECAM Bruxelles

Charles VANDEVOORDE (13019)
Lorenzo RIGA (13018)
Antoine VANDER MEIREN (12088)
Gaetano GIORDANO (12054)
Sylvain ALONSO (12150)

April 30, 2017

1 Introduction

Dans le cadre du cours d'applications mobiles, nous avons réalisé une application Android "ECalendar" à l'aide d'Android Studio. Cette dernière utilise l'API de "Ecam Calendar" afin de permettre aux étudiants de consulter leurs horaires directement sur leurs GSM.

Dans ce rapport, nous allons présenter la manière dont nous nous sommes organisés en groupe mais aussi la structure de notre application ainsi qu'un listing de ses fonctionnalités.

2 Organisation du groupe

Nous avons hébergé notre projet sur Github de manière à faciliter le travail en groupe et nous sommes répartis les tâches comme suit lors de la première séance:

Charles chargement des données depuis l'API, DAO ainsi que gestion de la *RecyclerView*.

Lorenzo gestion du layout, style de l'application.

Antoine

Gaetano gestion du layout et style de l'application

Sylvain intégration de la base de données SQLite et gestion des clics (affichage des détails lorsque l'utilisateur appuie sur un élément).

3 Présentation de l'architecture

3.1 Récupération des données depuis l'API REST

Notre application pour fonctionner a besoin de données provenant de l'API REST du site `calendar.ecam.be`. Après une analyse des requêtes faites par l'application web Calendar, nous avons tous les outils en main pour démarrer l'implémentation de l'API dans notre application.

Pour éviter de bloquer l'interface utilisateur pendant le chargement des données, nous avons utilisé un *IntentService*. Celui-ci est un service spécial d'Android qui s'arrête dès que la tâche est finie. Nous avons choisi ce type de service à la place de système comme les *AsyncTaskLoader*, ... pour plusieurs raisons.

- Meilleure séparation des affaires puisqu'un *IntentService* ne doit pas obligatoirement être appelé de l'activité principale.
- API plus propre grâce à l'utilisation d'*Intent*.

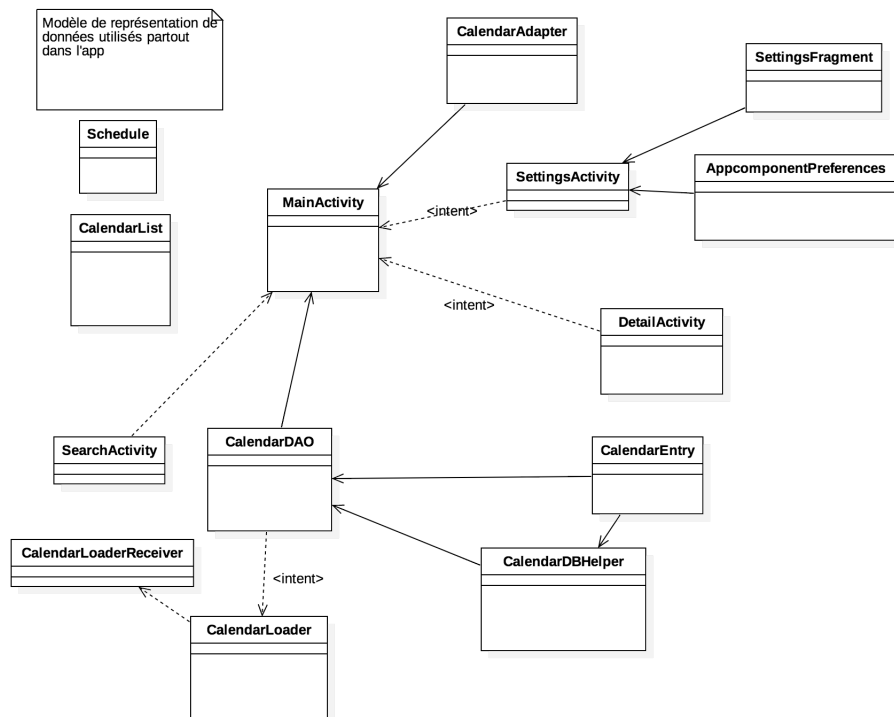


Figure 1: Diagramme UML de classe représentant notre application

Notre service peut être appelé pour récupérer deux types de données: la liste des calendriers disponibles (profs, locaux, séries, ...) et un calendrier. Pour chaque type, nous envoyons la requête via un *Intent* contenant les données nécessaires à la récupération des données.

Les requêtes HTTP sont faites synchrones à l'aide de *future* puisqu'Android ne supporte pas le multi-threading imbriqué. Pour ce faire, nous avons utilisé la librairie *Volley* qui est une nouvelle librairie permettant de faciliter les requêtes HTTP.

L'utilisation des *futures* vient du fait que *Volley* est asynchrone de base.

Le parsing des données ICS (format de donnée de calendrier) est effectué par une librairie appelée *iCal4j*. Le parsing est effectué dans l'*IntentService* pour éviter des ralentissements inutiles de l'interface utilisateur.

Finalement, les données sont renvoyées via un *Broadcast* jusqu'à un récepteur spécialement écrit pour ce service. Nous détaillerons plus en détail ce récepteur dans la section 3.1.

3.2 Le DAO

DAO signifie *Data Access Object*. Cette classe permet d'être le seul point d'accès aux données et décide de quel source les données seront tirées. Dans notre cas, les sources sont la base de donnée et l'API REST du site `calendar.ecam.be`.

La sélection de la source est assez trivial, nous regardons d'abord si les données sont en base de données et si elles sont assez nouvelles (moins de 1 semaines) aussi non, nous allons utiliser notre *IntentService* pour charger nos données et les sauver dans la base de donnée.

Mise à part la création, le DAO est responsable de la sauvegarde et de la reprise de données en base de données.

Le DAO devant gérer l'asynchronisme de l'*IntentService*, nous avons utilisé un système de "notifieurs" qui demande d'être notifié dès qu'une donnée est arrivée. Le DAO garde donc une liste de classe implémentant une certaine interface.

Une autre caractéristique du DAO est que celui-ci est implémenté comme un singleton parce que nous devons avoir le contexte d'une activité pour sauvegarder les données en base de données. Lorsque le *CalendarLoaderReceiver* (le récepteur de l'*IntentService*) reçoit les données, nous devons accéder au DAO mais le récepteur n'a pas de contexte. Il faut donc avoir un contexte statique (mauvaise pratique en général) ou alors implémenter le DAO entant que singleton. Cette décision nous a permis également d'être beaucoup plus flexible quand au chargement des données.

3.3 Le *RecyclerView*

Le *RecyclerView* nous permet d'afficher une grande liste de calendriers sans problème de performances. Dans notre cas, nous avons créé une *RecyclerView* avec deux types de données. Nous pouvions soit avoir un slot horaire, soit avoir une information sur la date des slots horaires suivant.

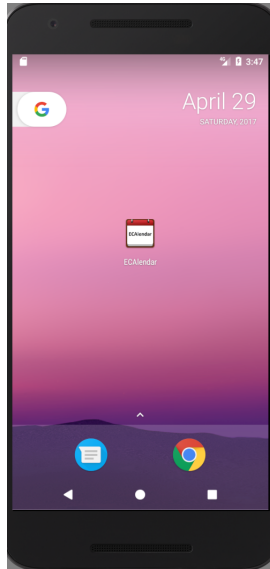
Dans notre *ItemAdapter* qui gère la création à la volée des informations affichées, nous avons donc deux différents *ViewHolder*.

La *RecyclerView* ne supportant qu'une seule liste de donnée, nous avons créer une liste d'*Object*. Cette approche pourrait être changé dans le futur si nous rajoutons plusieurs types de vues différentes. Dans notre cas, cette solution est la plus facile et la plus rapide à implémenter.

4 Listing des fonctionnalités

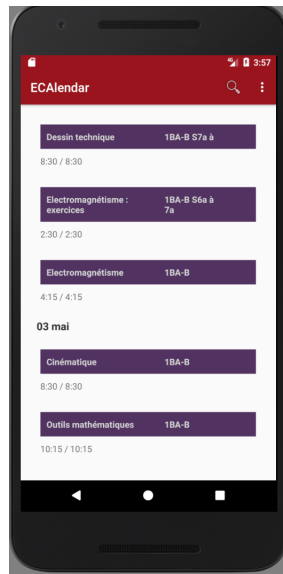
4.1 Icon

Que serait une application mobile, sans une icon appropriée. C'est pour cela que nous avons crée une icon ECAalendar spécialement customisée pour notre application.



4.2 La page d'accueil

Comme vous pouvez le voir sur l'image, en entrant dans l'application ECAalendar nous avons une page qui nous indique les cours pour l'année sélectionnée. Autrement dit elle indique la date ainsi que les cours donnés. Elle indique également l'heure du debut et l'heure de fin du cours.



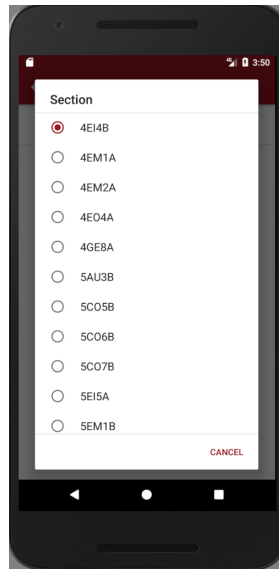
4.3 Détail d'un cours

En cliquant sur l'un des cours on entre dans une nouvelle page qui nous permet d'afficher les détails concernant ce cours. C'est à dire le local, l'enseignant et la série.



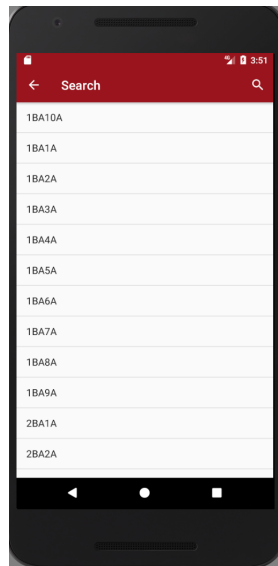
4.4 Choix de l'année

Bien évidemment, notre application nous permet de choisir également l'année d'étude pour lequel nous aimerions voir l'horaire. C'est pour cela que sur la page d'accueil en cliquant sur les 3 points en haute à droite, cela nous redirigera vers une page qui nous permettra de cocher une année et ainsi la mettre dans nos préférences.



4.5 Recherche

Nous pouvons également choisir l'année sur la page d'accueil, en cliquant simplement sur la petite loupe située en haut. Cela nous redirigera vers la page ci-dessous.



5 Conclusion