



---

GRAND CERCLE MOBILE - GCM

# Document d'implantation

Luiza CICONE - Jérémy KREIN - Jérémy LUQUET - Paul MAYER  
ISI - IF

---

Mai 2012

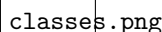


## Table des matières

1	Modèle objet	2
2	Présentation technologie	2

## 1 Modèle objet

Le diagramme suivant représente la modélisation objet du domaine d'application qui guide nos choix de conception.



classes.png

## 2 Présentation technologie

La technologie android repose sur la dualité entre la langage Java et le langage xml. A chaque affichage correspond un fichier xml qui met en place les différentes fenêtres qui constituent l'écran, appelées des « layouts », ainsi qu'un fichier Java qui représente ce qu'on appelle une activité, c'est à dire la fenêtre visible à l'écran. La vue offerte à l'utilisateur est mise en place par la méthode `setContentView([fichier].xml)`. Néanmoins, ce fichier xml en paramètre de la méthode mentionnée correspond à une configuration statique de l'affichage. Ainsi, dès que nous avons besoin de modifier dynamiquement une vue, nous le faisons dans le code Java, grâce à l'identifiant de la vue considérée renseigné dans le code xml : c'est ce qui fait la dualité Java/xml. Néanmoins, le placement des différents objets dans les layouts est moins facile en Java que dans un fichier xml, c'est pourquoi nous sommes passé par les fichiers xml dès que cela était possible.

Dans la technologie android, c'est un fichier appelé `AndroidManifest.xml` qui gère les différentes activités de l'application. La première activité lancée est celle contenue dans un label « intent-filter ». Dans notre cas, c'est le fichier `GCMLaunching.java` qui constitue l'activité de démarrage. Ce dernier fait appel à la méthode `parseFiles(Context context)`, qui utilise une classe factory pour obtenir une instance d'un `SAXParser`, nécessaire pour la récupération des données. En effet, l'application réalisée ne peut exister sans que le site du Grand Cercle ne soit tenu à jour. Nous récupérons ainsi toutes les données présentes sur ce site à l'aide de cinq parser, appelés par la méthode `parseFiles(Context context)` sur cinq fichiers xml différents en même temps que l'affichage d'accueil à l'aide de threads.

Plusieurs choses sont ainsi réalisées lors de cette phase :

- récupération des données sur le site
- création des listes de clubs, de cercles et de types d'événements, nécessaire pour la gestion des préférences
- initialisation de la base de données nécessaire pour la gestion des préférences

Un des points importants de la récupération de donnée est la présence ou non de connexion internet :

- soit une connexion internet est disponible. Dans ce cas on sauvegarde les fichiers xml en provenance du site dans la mémoire et on parse ces données.
- soit aucune connexion n'est possible et dans ce cas, on parse les données qui sont dans la mémoire, sauvegardées lors de connexions antérieures.

Lors de la première utilisation de l'application, un message est affiché à l'utilisateur s'il n'est pas connecté à internet pour lui indiquer qu'aucune donnée n'a pu être chargée (mais l'application se lance correctement). Lors de l'initialisation de la base de données, aucun filtre n'est appliqué et les données sont récupérées dans leur totalité, l'utilisateur ayant ensuite le choix d'appliquer ou non ces filtres.

Une fois la récupération des données et les initialisations effectuées, l'activité `GCM.java` est lancée. La différence avec les autres activités est qu'elle étend la classe `TabActivity` et non `Activity`. En effet, cette activité permet de construire les onglets de notre application à l'aide d'un objet appelé `TabHost`. Toutes les vues des différents onglets sont créés dans `GCM.java`, ce qui permet à l'utilisateur de naviguer rapidement entre les



différents onglets. Au sein de ces onglets, un clic sur les différents boutons lancent d'autres activités à l'aide de listeners