



T E A M U P

MANUEL DE MAINTENANCE

PROJET DE L'ÉQUIPE 3

HALLOUCHERIE LUCIE | LENESTOUR AÏNHOA | VIERAT NATHAN | RIVAS HUGO

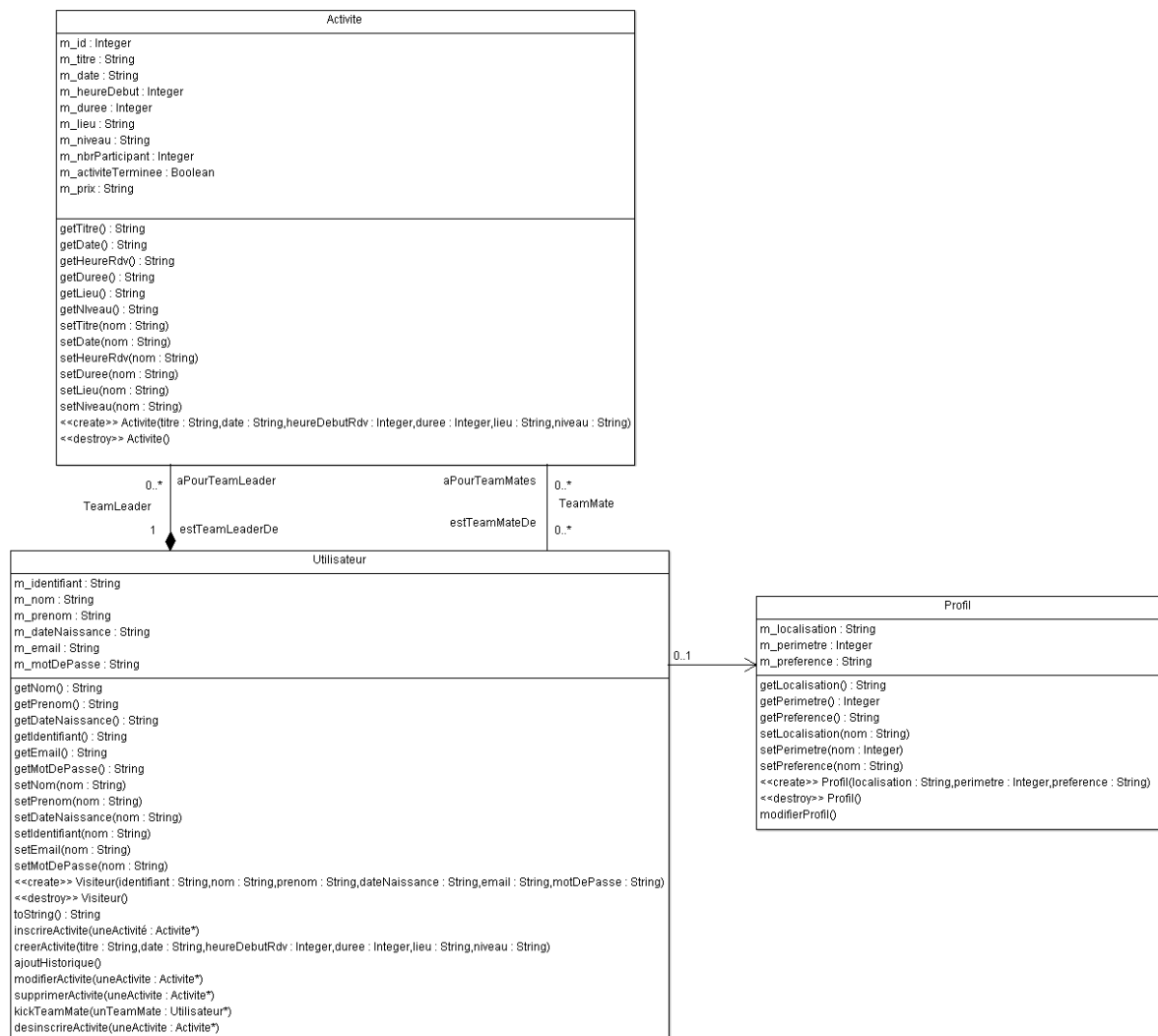
PROFESSEUR TUTEUR ET COMMANDITAIRE : MME. PANTXIKA DAGORRET
DUT INFORMATIQUE | ANNÉE UNIVERSITAIRE 2020-2022

Table des Matières

1.	Documentation détaillée du code et des modules	3
1.1.	Diagramme de classes.....	3
1.2.	Architecture du code	4
2.	Procédure de sauvegarde (système et données)	5
2.1.	Base de données & API	5
2.2.	Données locales	5
3.	Procédure de restauration.....	5
3.1.	Base de données & API	5
3.2.	Données locales	5
4.	Estimation de l'évolution du volume des données.....	6

1. Documentation détaillée du code et des modules

1.1. Diagramme de classes



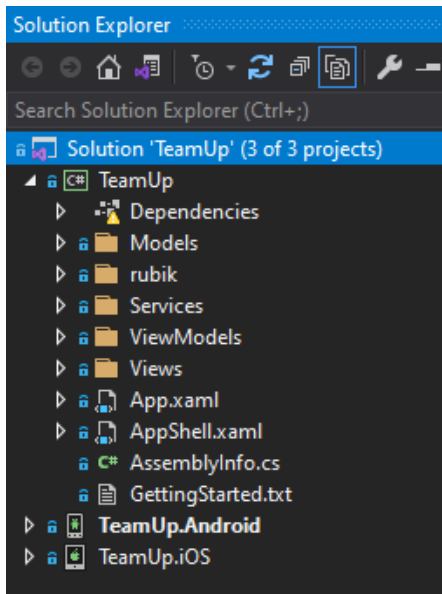
Le diagramme de classes ci-dessus représente la structure des classes et leurs méthodes, puis leurs relations avec les autres classes. On retrouve les classes :

- **Activité** : elle représente les activités de l'application, et possède deux relations avec la classe Utilisateur. La relation *aPourTeamLeader* fait référence à l'Utilisateur ayant créé l'Activité. La relation *aPourTeamMates* fait référence aux Utilisateurs inscrits à l'Activité en tant que Teammates.
- **Utilisateur** : elle représente les Utilisateurs dans l'application, et possède les mêmes relations avec Activité par les liens *estTeamLeaderDe* et *estTeamMateDe*, respectivement. Un Profil se crée dès l'apparition d'un Utilisateur et se voit relié à celui-ci.
- **Profil** : elle hérite de la classe Utilisateur et la complète avec des données dans le Profil de l'application. Elle est dépendante de la classe qu'elle hérite et se crée/supprime selon son père.

1.2. Architecture du code

La structure d'une application sur Xamarin.Forms est simple :

- Un dossier principal *TeamUp*, où les classes, vues, dépendances et d'autres fichiers pour générer l'application de base, donnent lieu à la création du code qui sera adapté selon l'OS.
- Un dossier *TeamUp.Android*, qui présente l'application en format Android.
- Un dossier *TeamUp.iOS*, qui présente l'application en format iOS (Apple).



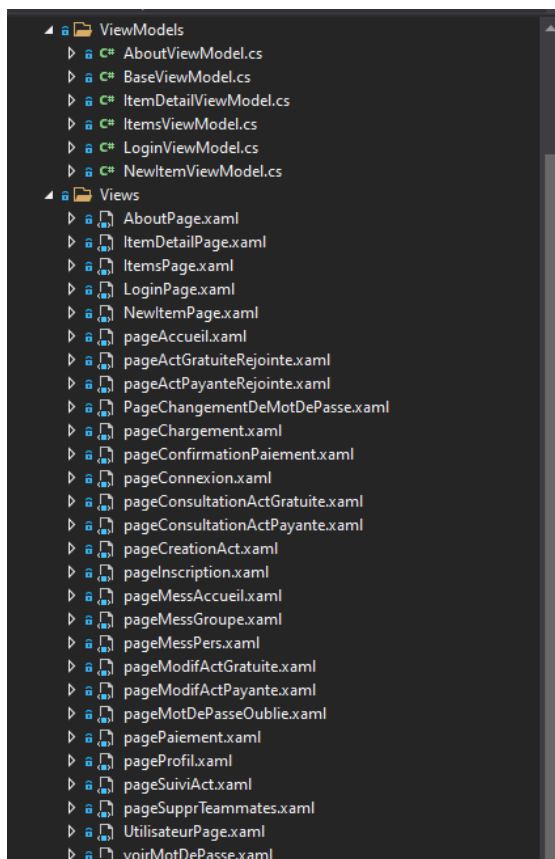
- *Dependencies* : paquets et librairies de l'application.

- *Models* : Les classes de modèle sont des classes non visuelles qui encapsulent les données de l'application.

- *Rubik* : il s'agit d'une librairie de polices.

- *Services* : fichiers qui servent à l'implémentation du code sur Android et iOS.

- *App.xaml* : c'est le constructeur principal de l'application, il s'initialise quand on lance Team Up.



- *ViewModels* : Le modèle de vue implémente les propriétés et les commandes auxquelles la vue peut être liée, et notifie la vue de toute modification d'état via des événements de notification de modification.

- *Views* : Les vues chargées de définir la structure, la disposition et l'apparence de ce que l'utilisateur voit sur l'écran (donc chaque page de l'application).

2. Procédure de sauvegarde (système et données)

2.1. Base de données & API

Au niveau de la sauvegarde des données, on utilise une base de données à distance depuis un serveur qui nous a été prêté. Pour la manipuler (consultation, ajout, modification et suppression de données), nous utilisons une API REST qu'on a créée, qui est disponible en ligne depuis le serveur.

Avec notre API, on peut sauvegarder toutes les modifications faites de nos données avec les méthodes courantes d'un REST (GET, POST, PUT, DELETE), qui s'occupera de modifier la base de données directement. De plus, l'API présente dans le serveur est un clone d'un repository GitHub du code original de l'API.

Cela nous facilite alors, lors d'un ajout de code ou correction de bug, l'actualisation de l'API en faisant un *git pull* du projet depuis la console de Git.

2.2. Données locales

Au niveau des données locales de l'application, on retrouve tous les icones et logos *Team Up* sauvegardés en dur et qui ne bougent pas, puis les informations de l'utilisateur actuellement connecté dans l'application. Ces dernières ne seront pas sauvegardées suite à la fermeture de l'application, car elles ne seront plus nécessaires dans la prochaine ouverture de Team Up.

3. Procédure de restauration

3.1. Base de données & API

Lorsqu'on lance l'application pour la première fois (ou après l'avoir fermée complètement), les données de note BD sont appelées depuis l'API.

3.2. Données locales

Les données locales sont forcément enregistrées en dur et elles sont chargées en parallèle du lancement de l'application.

4. Estimation de l'évolution du volume des données

L'évolution du volume des données de Team Up se reflètera surtout au niveau de la base de données à distance, car c'est celle-ci qui connaîtra une vraie augmentation de sa taille, au fur et à mesure que des nouveaux utilisateurs et activités se créeront.

Les données locales de l'application ne changeront pas, à moins qu'il y ait une actualisation à télécharger (bugs réglés, nouvelles fonctionnalités ajoutées...). Pour l'instant, l'APK de Team Up pèse environ 32 Mo.