

# Programmation des composants mobiles

## PROJET

### 1 Modalités

Le projet est à réaliser en binôme (ou, éventuellement, en monôme).

Les soutenances auront lieu soit en décembre soit en janvier, en fonction des dates disponibles (janvier semble plus probable). La date précise sera donnée ultérieurement.


Nous vous rappelons que chaque membre du binôme doit faire une partie conséquente du projet. Si, par exemple, l'un(e) d'entre vous ne fait que les layouts, il (elle) ne sera pas considéré(e) comme ayant fait une partie conséquente.

Pendant la soutenance chaque membre du binôme doit montrer la maîtrise de la totalité du code. Il est possible qu'une partie de la soutenance soit individuelle.

Par ailleurs, nous pourrions utiliser un ou des détecteurs de plagiat pour repérer les copies et autres "emprunts".

Vous devez créer pour votre projet un dépôt git sur gaufre


[https://gaufre.informatique.univ-paris-diderot.fr/users/sign\\_in](https://gaufre.informatique.univ-paris-diderot.fr/users/sign_in)

 avant le 20 novembre.

Les deux enseignants de l'UE doivent être désignés comme **Reporter** de votre projet sur git. Pour stocker votre projet Android dans dépôt git on procède de façon suivante :

- (1) Vous créez un projet sur AndroidStudio.
- (2) Sur AndroidStudio vous allez vers VCS -> **Enable Version Control Integration**.
- (3) **Select a version control system to associate with the project** : choisir Git.

Les noms de fichiers à sauvegarder dans git sont désormais coloriés en rouge.

-  (4) Dans le menu de AndroidStudio apparaît un onglet **Git** à la place de VCS. Allez vers **Git -> Manage Remotes**

Dans la fenêtre qui s'ouvre faire un clic sur +. Une autre fenêtre s'ouvre où vous copiez le lien https vers votre dépôt git.


- (5) L'étape suivante est peut-être inutile si votre nom et mail Git sont déjà renseignés globalement sur votre machine.

Vous allez vers le répertoire de votre projet et vous exécutez sur le terminal deux commandes :

```
git config user.name your_user_name
git config user.email your_email
```

**your\_user\_name** c'est évidemment votre nom sur Gitlab écrit entre les guillemets doubles. Vous pouvez voir ce nom en vous connectant sur le Gitlab gaufre et en allant vers l'onglet le plus à droite et en sélectionnant **Settings** (juste avant **Sign out**). D'habitude c'est votre nom et prénom avec l'espace entre les deux.

**your\_mail** c'est votre mail sur Gitlab gaufre, visible également dans **Settings**.

-  (6) Dans AndroidStudio vous allez vers **Git -> Commit**. Une sous-fenêtre s'ouvre à gauche avec tous les fichiers que AndroidStudio propose de sauvegarder dans le dépôt. Sélectionner tout. Écrire un **Commit message** et faire **Commit** ou directement **Commit and push**. Notez que commit est impossible si vous n'avez pas écrit quelque chose dans la sous-fenêtre Commit message.

Bien sûr votre binôme doit plutôt commencer par cloner le projet.

Dans le dépôt git vous ajoutez `README.md` où vous mettrez les noms de tous les membres de votre équipe de projet.

→ Le projet doit être réalisé en Kotlin.

( Dans le projet, il y aura sans doute des listes. **Toutes les listes affichées doivent être implémentées par des RecyclerViews** avec des **Adapters** appropriés.

( L'utilisation de fragments n'est pas demandée mais les fragments peuvent apporter des points s'ils sont bien faits.

L'utilisation de `ViewModels` dans toutes les activités qui possèdent les données à préserver pendant le changement de configuration est très très recommandée.

Nous allons vérifier pendant la soutenance le comportement de votre application durant le changement de position de l'appareil. Si dans votre application vous interdisez de faire tourner l'écran cela enlèvera des points à votre évaluation.

## Présentation de projet

C'est un projet qui est sensé aider un amateur (ou une amatrice) de fleurs et d'autres plantes à les maintenir en vie. Son problème c'est l'oubli d'arrosage pour certaines plantes et/ou l'arrosage trop fréquent pour d'autres.

Il lui faut une application qui fait les rappels à des moments appropriés.

On peut aussi introduire la distinction entre un arrosage simple et un arrosage avec des nutriments.

## Base de données

Vous allez maintenir une base de données. Pour chaque plante on stocke son nom, ou même deux noms — le nom commun et le nom latin — au moins un des deux noms doit être non-vide.

Pour chaque plante on maintient des informations sur la fréquence de l'arrosage. En fait il s'agit plutôt de plusieurs fréquences<sup>1</sup> parce que pour certaines plantes la fréquence peut dépendre de la saison, genre deux fois par semaine entre de mai et septembre et une fois chaque 10 jours tous les autres mois. Si vous distinguez l'arrosage simple et l'arrosage avec les nutriments il faut aussi des fréquences d'arrosage nutriment qui sont des multiples de fréquences d'arrosage simple. Par exemple si l'arrosage simple se fait une fois chaque 10 jours alors l'arrosage avec les nutriments une fois chaque 30 jours.

Le format exact de stockage de ces données dans la BD est à déterminer (par vous, pas par moi). Ce qui compte c'est la simplicité, et la facilité d'implémentation des activités/-services qui utilisent la BD.

Vous stockez dans la BD toute information pertinente, par exemple la date du dernier arrosage de chaque plante ou la date du prochain arrosage (ou peut-être les deux si cela est utile).

---

1. disons 3 au maximum

Pour chaque plante on stocke aussi un lien vers la photo de la plante (photo prise à l'aide de l'appareil, ou copiée comme un fichier depuis un ordinateur vers la mémoire de l'appareil ou téléchargée sur internet, on peut envisager différentes sources de photo).

## 2 Activité pour ajouter une nouvelle plante

Évidemment il faut une activité permettant d'ajouter une nouvelle plante.

## 3 Activité de suppression/modification de données

Il faut aussi pouvoir modifier les données sur une plante et supprimer une plante (qui s'est desséchée malgré nos efforts).

## 4 Parcourir/rechercher/afficher les plantes

Il est nécessaire d'avoir la possibilité de consulter la BD, afficher les plantes avec leurs photos et avec les périodes/fréquences d'arrosage. Cela peut s'accompagner de recherche de plantes par le nom, de la possibilité de parcourir toutes les fiches de plantes.

## 5 Alarme : le coeur de l'application

L'application doit posséder une activité qui permet d'installer une alarme qui se déclenche périodiquement chaque 24h.

L'alarme fait activer un service qui consulte la BD pour voir quels arrosages faut-il effectuer le jour donnée. Cela implique, bien sûr, que dans la BD on garde pour chaque plante la date du dernier arrosage (ou la date du prochain arrosage). La date du dernier arrosage avec la périodicité de l'arrosage permettent de calculer si le jour courant est le jour d'arrosage. Si on décide plutôt de garder la date du prochain arrosage alors en utilisant la périodicité l'application et cette date on pourra calculer la date de suivant arrosage.

### 5.1 Notification et l'activité d'arrosage

S'il y a des arrosages à effectuer le service installe une notification. Quand l'utilisateur fait un clic sur la notification on ouvre une activité d'arrosage qui nous informe quelles plantes sont à arroser. Il est possible d'ouvrir cette activité d'arrosage sans attendre la notification.

L'activité d'arrosage donne seulement les consignes d'arrosage (hélas elle ne fait pas encore d'arrosage elle-même). L'utilisateur n'est pas tenu à suivre ces consignes, par exemple il peut constater que le sol d'une plante est encore humide et décider de ne pas l'arroser ce jour.

Parmi les plantes à arroser l'utilisateur marque celles qu'il n'a pas arrosé. Pour ces plantes on peut envisager deux comportements, soit l'arrosage non-effectué est automatiquement re-planifié pour le jour suivant, soit avec l'aide de `DatePickerDialog` l'utilisateur choisit la date du prochain arrosage.