



Rapport de TP - MyNotes

Application Android MyNotes v1.0

BENCHINE Abdelilah - GINF3
17/12/2025

1. Introduction

Dans le cadre du TP2 de développement mobile Android, nous avons réalisé une application native nommée **MyNotes v1.0**.

Cette application permet à l'utilisateur de gérer ses notes personnelles de manière simple et organisée à travers une interface intuitive.

L'objectif principal de ce TP est de maîtriser :

- la création de plusieurs activités Android,
- la navigation entre écrans à l'aide des Intents,
- l'utilisation d'un **Adapter personnalisé** pour l'affichage de données,
- et la gestion du cycle de vie des activités.

2. Description Fonctionnelle de l'Application

L'application **MyNotes** offre les fonctionnalités suivantes :

2.1 Visualisation des notes

- L'écran principal affiche la **liste de toutes les notes** enregistrées.
- Chaque note est affichée avec :
 - son **nom (titre)**,
 - sa **date**,
 - une **couleur distinctive** représentant le niveau de priorité.

2.2 Ajout d'une nouvelle note

- Un bouton « + » permet d'accéder à un formulaire d'ajout.
- L'utilisateur saisit :
 - le nom de la note,
 - sa description,
 - la date,
 - la priorité (Basse, Moyenne ou Haute).
- Après validation, la note est ajoutée à la liste et l'application revient automatiquement à l'écran principal.

2.3 Consultation des détails d'une note

- En cliquant sur une note depuis la liste, l'utilisateur accède à un écran de détails.
- Cet écran affiche l'ensemble des informations de la note sélectionnée.
- Un bouton « Retour » permet de revenir à la liste des notes.

3. Structure des Données

Chaque note est représentée par une classe Java **Note** contenant les attributs suivants :

- **nom** : titre de la note,
- **description** : contenu détaillé,

-
- **date** : date de création ou d'échéance,
 - **priorité** : niveau d'importance (Basse, Moyenne, Haute).

Cette structure permet une manipulation simple et efficace des données dans l'application.

4. Architecture de l'Application

L'application est composée de trois activités principales :

- **NoteListActivity**
Affiche la liste des notes à l'aide d'une **ListView** et d'un adaptateur personnalisé.
- **AddNoteActivity**
Permet la saisie et l'enregistrement d'une nouvelle note via un formulaire.
- **DetailsNoteActivity**
Affiche le détail complet d'une note sélectionnée.

La navigation entre ces activités est assurée par des **Intents explicites**.

5. Description de l'Adapter Personnalisé

Pour afficher les notes dans la **ListView**, un **ArrayAdapter personnalisé** nommé **NoteAdapter** a été implémenté.

5.1 Rôle de l'Adapter

L'adapter a pour rôle de :

- convertir chaque objet **Note** en une vue graphique,
- afficher plusieurs informations sur une même ligne (nom et date),

-
- appliquer une couleur spécifique selon la priorité de la note.

5.2 Fonctionnement

- La méthode `getView()` est redéfinie afin de charger un layout personnalisé (`row_note.xml`).
- Chaque ligne contient :
 - un `TextView` pour le nom,
 - un `TextView` pour la date,
 - une barre colorée indiquant la priorité.
- La couleur est choisie dynamiquement :
 - **Vert** pour une priorité basse,
 - **Orange** pour une priorité moyenne,
 - **Rouge** pour une priorité haute.

Ce mécanisme améliore la lisibilité et l'ergonomie de l'application.

6. Description de l'Adapter Personnalisé

- Lors de l'ajout d'une note, l'activité `AddNoteActivity` renvoie les données à l'activité principale grâce à `setResult()` puis se ferme avec `finish()`.
- L'activité `NoteListActivity` récupère ces données et met à jour la liste.
- Le retour depuis l'écran de détails est également géré via `finish()`,

garantissant une navigation fluide.

Conclusion

L'application **MyNotes v1.0** répond aux exigences du cahier des charges du TP2. Elle met en œuvre les concepts fondamentaux du développement Android, notamment :

- la gestion de plusieurs activités,
- la communication entre écrans,
- l'utilisation d'un adaptateur personnalisé,
- et une interface utilisateur claire et ergonomique.

Ce projet constitue une base solide pour des évolutions futures telles que la persistance des données, l'ajout de photos ou l'amélioration de l'interface utilisateur.

