

HMIN205 - Programmation Mobile

TP N°3

PERSISTANCE DE DONNEES

Tous les exercices de ce TP sont indépendants. Cependant, ils partagent les interfaces (Views) utilisées.

Exercice 1 : Sauvegarde de l'état d'une activité

Créez une application constituée d'une activité pour l'inscription de l'utilisateur, activité *Inscription*. Cette activité permet à l'utilisateur de saisir les informations suivantes : *nom*, *prénom*, *âge*, *numéro de téléphone* et un *mot de passe*.

- Faites pour que toutes les informations saisies du formulaire, sauf le mot de passe, soient sauvegardées quelque que soit l'état de l'activité (par exemple, passage de l'application du mode portrait au mode paysage).
- Chaque utilisateur possède également un identifiant généré aléatoirement par une méthode de l'activité d'inscription. Définissez la méthode de génération de cet identifiant et faites pour que cet identifiant soit sauvegardé quelque que soit l'état de l'activité.

Exercice 2 : Manipulation de fichiers

Considérez l'application créée dans l'exercice précédent.

- Sauvegardez les informations d'un utilisateur dans un fichier dont le nom est le résultat de la concaténation du *nom* de l'utilisateur et de son *identifiant*.
- Ajoutez un bouton (bouton « *Soumettre* ») à l'activité d'inscription. L'activation de ce bouton permet de lancer une deuxième activité qui affiche les informations sauvegardées dans le fichier précédent.

Exercice 3 : LifeCycle

Reprenez l'application définie lors des exercices précédents.

- Ajoutez à cette application une classe sensible au Cycle de vie de l'activité (Lifecycle Aware), classe *Utilisation*. Cette classe contient une méthode dont le nom est *NombreUtilisation()*. Cette méthode compte le nombre de fois où l'activité a été relancée (suite à l'appel de la méthode *OnResume()*).
- Faites que ce nombre soit affiché avec les autres informations suite à l'activation du bouton « *Soumettre* ».

Exercice 4 : ViewModel

Reprenez l'application définie lors des exercices précédents.

- Ajoutez une activité nommée « *planning* » qui affiche le planning journalier de l'utilisateur. Ce planning contient le programme de l'utilisateur pendant la journée en cours. Une journée est découpée en 4 créneaux : *08h-10h*, *10h-12h*, *14h-16h* et *16h-18h*.

Par exemple, l'activité affiche :

08h-10h : Rencontre client Dupont

10h-12h : Travailler le dossier recrutement

14h-16h : Réunion équipe

16h-18h : Préparation dossier vente.

- Cette activité se lance par un deuxième bouton placé sur l'activité d'inscription : bouton « *Affiche Planning* ».
- Ces informations sont disponibles comme les valeurs des attributs d'une classe *ViewModel* associée à l'activité en question (classe *PlanningModel*).

Exercice 5 : LiveData

Reprenez l'application définie lors des exercices précédents.

- Faites que les informations du planning détenues par les attributs de la classe *PlanningModel* soient mises à jour, au moins, une fois par jour, suite à la lecture d'un fichier qui stocke ces informations (fichier « *planning* »).
- Utilisez le concept de *LiveData* pour rafraichir le planning affichée suite à la mise à jour de ce dernier dans le *PlanningModel*.

Exercice 6 : Base de données Room

Reprenez l'application définie lors des exercices précédents.

- Sauvegardez les informations précédentes dans une base de données Room : Créez les entités, les DAO et la Base de données.
- Ajoutez à la classe *PlanningModel* une méthode qui permet de mettre à jour les attributs de cette classe à partir des données de la base de données.