

Projet universitaire

- BERTRAND Pierre-Louis
- NEONAKIS Ionas
- TOULARHMINE Samir

Documentation technique : onTime

28 novembre 2020

Vue d'ensemble

Dans ce document vous trouverez une documentation technique de l'application Android onTime.

Objectifs

1. **Architecture du code**
2. **Explication des fonctionnalités**

Grandes étapes

1. Architecture du code

Tout d'abord le code est répertorié sous plusieurs packages Java permettant une meilleure organisation des fichiers.

- Dans le package "activities" vous trouverez les différentes Activity utilisées par l'application.

- Dans le package “adapters” se trouvent les différents Adapters utilisés par les RecyclerView.
- Dans le package “fragments” se trouvent les Fragments utilisés par l’application.
- Le package “items_touch_helpers” contient les ItemTouchHelpers utilisés dans les RecyclerView pour la gestion du Swipe et du drag’n drop.
- Vous trouverez dans le package “layouts” le layout principal de l’application réalisé par nos soins.
- Dans le package “modele” se trouvent les classes structurant les données utilisées par notre application.
- Le package “service” contient les classes pouvant gérer les APIs.

Nous avons fait le choix de baser notre application sur les fragments car nous avons estimé que l’expérience utilisateur était meilleure. La navigation dans l’application se fait donc principalement par la barre de navigation se situant en bas.

- **Barre de navigation**

Les éléments de la barre de navigation sont définis dans le fichier “bottom_navigation_menu.xml”. Vous y trouverez les onglets déjà existants, chacun avec un ID et un titre. Cet ID nous permettra de référencer l’onglet du menu depuis le fichier “nav_graph.xml” où sont définis les différentes transition possible d’un fragment à un autre.

Ce graphe sera utilisé par le contrôleur de la barre de navigation dans le fichier “HomeActivity.java”.

- **Présentation des fragments**

HomeFragment : vue principale de l’application à l’écran d’accueil où on y retrouve le TimePicker pour définir l’heure d’arrivée, le bouton pour mettre en place le prochain réveil, le fait de pouvoir sélectionner une morning routine, une liste des tâches à faire pendant cette dernière et un bouton de départ à la fin.

EditTFFragment : vue pour éditer un fragment avec une carte GMaps contenant les points d'arrivée et de destination du trajet ainsi qu'un champ de texte pour éditer le titre de ce trajet.

ListMRFFragment : vue listant les différentes Morning Routines de l'utilisateur représentées par une CardView avec à gauche le nom de la Morning Routine et à droite le trajet actuellement associé.

ListeTachesRecurrentesFragment : vue listant les tâches récurrentes que l'utilisateur a créées. Elles sont représentées par un CardView avec à gauche le nom de la tâche et à droite la durée. Il y a également la définition du bouton d'ajout ; ouvrant une popup pour créer la tâche récurrente.

ListTFFragment : vue listant les trajets de l'utilisateur. Ces derniers sont représentés par une CardView avec à gauche le titre, l'adresse de départ et l'adresse d'arrivée puis à droite un bouton d'édition du trajet renvoyant vers le fragment EditTFFragment.

ManageFragment : vue représentant l'onglet "Gérer" où l'on peut gérer les différentes données de l'utilisateur : ses Morning Routines, ses Trajets ainsi que ses tâches récurrentes.

SettingsFragment : vue représentant les paramètres utilisateurs. Il peut activer ou non les notifications avant chaque tâche, retourner au tutoriel inclus dans l'application ou encore supprimer ses données.

Remarque : Pour tous les fragments contenant des listes, il y a un adapter correspondant dans le package "adapter" et donc un item layout correspondant dans le dossier "layout" des ressources. Mais il y a aussi un ItemTouchHelper afin de gérer le swipe de suppression ainsi que le drag'n drop de l'élément.

Remarque : Les textes de l'application sont répertoriés dans le dossier "values" des ressources afin de permettre l'internationalisation de ces derniers. Actuellement, l'Anglais et le Français sont entièrement disponibles pour l'utilisateur. La langue se base sur le langage du système.

2. Présentation des fonctionnalités

- Sauvegarde des données

La sauvegarde des données de l'utilisateur se fait par l'utilisation de SharedPreferences dans le contexte de l'application. Ces dernières sont stockées sous le nom de "onTimePreferences".

Remarque : Certaines valeurs sont stockées de manière temporaire dans les sharedPreferences afin d'assurer la sauvegarde des données lors d'un changement de fragment par exemple. Par conséquent, elles sont supprimées après utilisation.

On y stocke :

- le fait que l'utilisateur aie fini ou non le tutoriel de premier démarrage
- la morning routine courante
- le gestionnaire de morning routine
- l'id de la morning routine sélectionnée
- la position du trajet sélectionné
- le trajet sélectionné
- l'id maximum des morning routines créées
- la liste des trajets
- la liste des tâches récurrentes
- le fait de notifier ou non l'utilisateur

Nom	Type	Temporaire
CurrentMRA	MRA	Non
current_id_MRA	int	Non
MRManager	MRManager	Non
userHasFinishedInitialSetup	boolean	Non

morning_routine	MorningRoutine	Oui
position	int	Oui
trajet	Trajet	Oui
id_max	int	Non
listeTrajets	List<Trajet>	Non
listeTachesRec	List<Tache>	Non
ridingMethod	int	Non
notifyOnEachTaskStart	boolean	Non

- Classe Toolbox

Dans cette classe, vous trouverez différentes méthodes utilitaires qui sont utiles pour certains calculs par exemple.

- Éléments de designs utilisés

Beaucoup d'éléments de l'application repose sur l'implémentation Material Design UI de Google. Nous avons réalisé des maquettes en amont pour avoir une idée de ce à quoi pourrait ressembler l'application. Ces maquettes sont disponibles sur le dépôt.

- Tutoriel

Le tutoriel a été réalisé en partie grâce à la librairie AppIntro. Quelques screens et une réflexion sur les textes à mettre ont été fait afin d'assembler ce tutoriel de premier démarrage.