

19/12/2021

# ANDROID iHaveToDo

PROJET INFO0306



android

# Sommaire

## Table des matières

Sommaire .....	1
Introduction.....	2
Fonctionnement de l'application .....	2
Explications .....	2
Fonctionnalités.....	3
Activité + Internationalisation de l'App.....	3
Ressources.....	3
Orientation et portabilité de l'écran .....	4
La notification.....	4
Menu dans la barre des statuts.....	4
sharedPreferences.....	4
Bundle : sauvegarde d'état de l'activité.....	5
Capteurs .....	5
Ecran tactile .....	5
Fragments.....	5
Microphone .....	6
Base de données embarquée SQLite.....	6
BroadcastReceiver .....	6
Conclusion .....	7

# Introduction

Mon projet, intitulé « iHaveToDo » consiste à développer une application de gestion des tâches pour les smartphones Android.

Le but étant de vous présenter un projet dans son état final, son fonctionnement et les fonctionnalités qu'il contient. Pour cela je vais vous présenter dans chaque cas une description et décrire son utilisation pour vous permettre de vous en servir sans connaître son fonctionnement interne.

## Nom de l'application : iHaveToDo

- Nous n'avons pas eu de précision concernant le rapport, sachant qu'en général un rapport ne doit pas contenir de code, j'ai essayé d'expliquer au mieux chaque partie de mon application.

# Fonctionnement de l'application

## Explications

Lors de votre arrivée sur l'application vous pouvez apercevoir 3 ensembles de listes déroulantes, qui répertorient les tâches que vous souhaitez effectuer en fonction de la date et l'horaire que vous entrez.

Pour la page où vous souhaitez ajouter une nouvelle tâche, vous aurez la possibilité de choisir la date et l'heure auxquelles vous devez être informé de cette tâche, de sorte que celle-ci soit effectuée avant d'être supprimée lorsque la date et l'horaire de celle-ci sera passée.

Ensuite, la tâche vous sera montrée sur la page d'accueil, et vous la trouverez triée à la bonne position dans le bon groupe pour un affichage clair et intuitif.

# Fonctionnalités

Dans le cadre de ce projet, nous avons certaines fonctionnalités à implémenter à notre application et qui devaient être en corrélation avec le thème choisie.

J'ai donc implémenté toutes les fonctionnalités présente ci-dessous. Pour chacune d'entre elles je vous explique son utilisation au sein de mon application.

Sans manquer d'avoir une architecture propre de mon dossier d'application pour permettre à une personne tierce de s'y retrouver facilement et lui permettre de trouver ce qu'il recherche sans perte de temps inutile.

## Activité + Internationalisation de l'App

Mon application contient au moins 2 activités. La page d'accueil où nous pouvons consulter la liste des tâches à réalisées et elles sont triés par groupe.

- Nous avons les tâches de notre journée actuelle : Aujourd'hui
- Les tâches qui sont à faire pour le lendemain : Demain
- Les tâches qui sont à venir : Plus Tard

Ainsi qu'une activité qui va nous permettre de créer ces tâches et d'indiquer l'heure et la date à laquelle elles doivent être réalisé.

Les langues qui sont incluse à l'application sont bien évidemment : l'anglais et le français. L'application est donc bien internationalisée avec le strict nécessaire. Les chaines sont déclarées dans le fichier **strings.xml** du répertoire ressource **res/values** utilisé pour une traduction en anglais et français.

## Ressources

Nous pouvons distinguer plusieurs types de ressources :

L'application fait usage d'image que l'on peut retrouver dans un répertoire : res  
Incluant une police d'écriture et des couleurs adaptés pour le confort de l'utilisateur durant son passage sur l'application. Nous pouvons y retrouver toute ressources de l'application telle que le design des fichiers **.xml** utilisé durant sa création.

## Orientation et portabilité de l'écran

iHaveToDo est une application totalement responsive et s'adapte à tout type d'écran et également à son sens d'orientation : portrait ou paysage.

Il sait s'adapter et positionner correctement le contenu visuel de l'application sur l'appareil de l'utilisateur pour lui permettre d'avoir un meilleur confort et une utilisation sans souci et très simpliste. Elle est implémentée depuis le layout xml qui utilise le scrollView Bar.

## La notification

Cette fonctionnalité importante permet de notifier l'utilisateur afin qu'il puisse accomplir sa tâche qu'il s'est donnée et ne pas l'omettre.

Elle permet également dans un sens d'éviter toute procrastination, une fois la notification reçue, ils ne peuvent l'ignorer et ne peuvent s'en prendre qu'à eux si celle-ci n'est pas faite car l'objectif ici est de prévenir l'utilisateur mais pas de contrôler si cela a bien été fait ou non. Nous ne jugeons point ce détail.

## Menu dans la barre des statuts

Le menu permet de naviguer entre les pages.

J'avais créé une page avec l'historique des tâches listées où l'on pouvait supprimer l'intégralité de nos tâches mais je l'ai remplacé par une suppression des tâches qui se fait d'une manière autonome, la tâche est désormais supprimée après que sa date soit passée. Elle affiche donc désormais la tâche mais sans la possibilité de supprimer manuellement. Le dossier **menu** est utilisé dans la barre d'état.

## sharedPreferences

Le projet inclut ce principe qu'est d'enregistrer des informations primitives de l'utilisateur selon ses préférences sur l'application.

Ceci est utilisé pour maintenir l'état de la liste quotidienne du jour, si l'on quitte puis revient sur l'application nous y revenons par défaut à la même position que la précédente si par exemple nous décidons de dérouler la liste à puces. L'état de l'application ne se réinitialise pas et garde cette trace en mémoire.

## Bundle : sauvegarde d'état de l'activité

Ce Bundle est utilisé lorsque nous programmons l'alarme pour un rappel. L'état est donc sauvegardé et est affiché tant que la tâche reste active.

Cela permet ainsi à l'utilisateur de consulter maintes fois les tâches qu'il se doit de réaliser et celles qui sont à venir. S'effectue lorsque l'on ajoute une nouvelle tâche (un nouveau rappel).

## Capteurs

J'utilise dans un premier temps un capteur de mouvement pour permettre d'orienter correctement l'affichage sur l'appareil de l'utilisateur.

Ainsi qu'un capteur sonore qui détecte le son environnant, cela récupère la source du bruit. Détecte donc le niveau sonore autour du téléphone à l'aide du microphone.

## Ecran tactile

Pouvoir supprimer une tâche de l'historique de manière à ce que seule celle-ci soit supprimée, cette action est effectuée lorsque l'on maintient la tâche ciblée et nous fait apparaître une pop-up afin de supprimer la tâche.

On programme une alarme, on quitte l'app, une fois la notification reçue nous nous trouvons sur une nouvelle page après avoir cliqué dessus où nous pourrions consulter cette historique. (si je retrouve la bonne version que j'avais à mettre dans le dossier pour la suppression unique).

## Fragments

L'utilisation du fragment a été utilisée dans une forme de vue avec son propre cycle, il est exécuté dans le contexte d'une activité.

Les fragments que j'ai utilisés se trouvent dans le répertoire `app/src/main` une fois dans ce package nous y trouvons plusieurs dossiers :

- Dossier java : pour trouver des fragments de notre app
- Dossier res (ressource) puis layout pour y trouver le layout associé

Chaque fragment est une partie de l'application, elles appellent la méthode **onCreateView** qui permet de récupérer le layout associé.

Par exemple rendre la liste à puces déroulante sur la page d'accueil est fait via l'utilisation d'un fragment.

## Microphone

L'utilisation du micro est faite lors de l'enregistrement d'une nouvelle tâche, nous pouvons donc la faire via synthèse vocale ou bien l'écrire à l'aide du clavier du smartphone directement pour écrire.

Lors de son utilisation le texte est automatiquement mis à jour au fur et à mesure puis entré dans le champ une fois un temps d'arrêt de la voix marqué par l'utilisateur.

La bibliothèque du système android est utilisé pour cette fonctionnalité **GetSpeech to text**.

## Base de données embarquée SQLite

J'ai fait usage de la base de données embarquée SQLite avec une utilisation de RoomDB qui va me permettre de stocker des données localement sous forme de chaîne de caractères, les informations stockées seront par exemple : les tâches enregistrer ainsi que leur date et heure si indiqué par l'utilisateur.

De cette manière je peux utiliser ces données pour les afficher ou bien vérifier la date pour effectuer la suppression de celle-ci.

## BroadcastReceiver

Le BroadcastReceiver permet l'utilisation de l'alarme pour un rappel des tâches qui se fait sur l'appareil de l'utilisateur.

Par conséquent, celui-ci émettra une alerte pour avertir l'utilisateur et lui rappeler les tâches dans lesquelles il s'est lancé.

Ceci est donc déclenché lorsque le temps d'une alarme est écoulé, ce qui déclenche un signal pour pouvoir afficher la liste des tâches.

## Conclusion

Pour conclure, tout au long de la préparation de notre projet, nous avons essayé de mettre en pratique les connaissances acquises durant ce semestre universitaires et cela dans le but de réaliser une application sur la gestion des tâches.

L'application est fonctionnelle bien que simpliste mais cela rend son utilisation intuitive du point de vue du consommateur qui va l'utiliser. Dans l'ensemble les objectifs fixés quant à la réalisation de mon application ont été majoritairement respectés.

Pour second objectif, j'aurais pu permettre à l'utilisateur de d'ajouter une localisation en plus de la tâche à faire ainsi que pouvoir communiquer avec un autre appareil en Bluetooth.

Je n'ai pas pu malheureusement intégrer la possibilité d'avoir des identifiants pour par la suite leur permettre d'interagir entre eux et de s'échanger des tâches communes. J'ai donc opté pour une solution ci-présente afin de vous proposer une application finalisée et fonctionnelle.

