

ÉCOLE POLYTECHNIQUE DE TUNISIE

RAPPORT DU PORJET: DÉVELOPPEMENT MOBILE



MALEK MECHERGUI

2019-2020

TABLE DES MATIÈRES

- 1. Présentation de l'application**
- 2. Vue générale sur l'application**
- 3. Description de l'application**
 - 3.1. Icône**
 - 3.2. Ecran de bienvenu**
 - 3.3. Ecran d'accueil**
 - 3.4. Rappel journalier et notification**
 - 3.5. Daily To Do List**
 - 3.6. Weekly Must Do List**
 - 3.7. Grocery Shopping List**

I. Présentation de l'application :

MyAgenda est une application mobile alternative plus pratique du typique agenda en papier. MyAgenda simplifie le quotidien en permettant à ses utilisateurs d'organiser leur temps. Elle donne à ses utilisateurs la possibilité de planifier, de noter leurs emplois du temps, leurs rendez-vous, leurs besoins, etc.

« Gagner du temps et tirer le maximum de votre journée » est le slogan de MyAgenda.

L'application MyAgenda est gratuitement téléchargeable et exécutable à partir de tout Smartphone doté d'un système d'exploitation Android. Elle est adaptée aux différents environnements techniques des Smartphones et à leurs contraintes et possibilités ergonomiques.

2. Vue générale sur l'application :

L'application MyAgenda se compose principalement de 3 compartiments à remplir quotidiennement et selon le besoin par son utilisateur. L'application ne nécessite pas une connexion internet et le contenu de ses compartiments reste enregistré même en sortant de l'application et ne s'efface que quand l'utilisateur appuie dessus. Les principales composantes de MyAgenda sont :

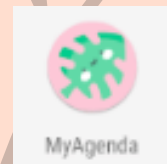
- La « Daily To Do List » ou les tâches journalières à faire : Ce compartiment est constitué de 4 listes à remplir chaque jour répartis selon la nature de la tâche. Les 4 listes sont en effet, les tâches/les objectifs que l'utilisateur doit accomplir durant sa journée ou les « Must Do », les tâches les moins importantes que l'utilisateur veut faire le long de sa journée « To Do », les appels que l'utilisateur doit faire « Calls To Make » et les endroits où l'utilisateur doit aller durant la journée « Places To Go ». L'application veille à ce que l'utilisateur remplisse chaque jour sa « Daily To Do List » en créant un rappel à une heure que l'utilisateur précise.
- La « Weekly Must Do List » ou les tâches à faire pendant la semaine : Réparti en 7 listes des 7 jours de la semaine. Ces listes peuvent être remplies en début de semaine pour que l'utilisateur ait une vue globale sur la semaine. Puisque ces listes contiennent des tâches importantes, chaque jour, les tâches qui correspondent à ce jour sont automatiquement transférées vers la liste des tâches que l'utilisateur doit faire ou les « Must Do » du compartiment des tâches journalières à faire ou la « Daily To Do List ».

- La « Grocery Shopping List » ou les courses à faire : pour minimiser les achats impulsifs, l'application est dotée de ce compartiment repartis en 9 listes, une liste par rayon (les fruits, les légumes, les produits laitiers, les viandes et les poissons, etc) et une liste supplémentaire au cas où le produit n'appartient à aucune des 8 listes prédéfinies.

3. Description de l'application :

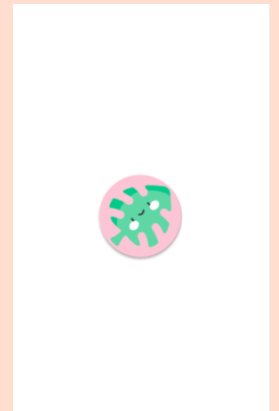
3.1. Icône :

Android Studio permet de personnaliser l'icône en créant un nouveau élément image (Image Asset) puis en chargeant une image de type *.png.



3.2. Ecran de bienvenu :

L'écran de bienvenu ou l'écran de déclenchement est un écran blanc qui contient généralement le logo de l'application et qu'on trouve dans plusieurs applications mobiles (Facebook, Instagram, YouTube, etc). Cet écran s'affiche le temps que l'application se charge. La création de cet écran se fait simplement par l'ajout d'une nouvelle classe Java (Welcome.java) dans le dossier java et un fichier xml welcome.xml dans le dossier drawable pour concevoir le design de cet écran. Il faut également faire quelques modifications dans le fichier manifest (AndroidManifest.xml) pour spécifier la catégorie de l'activité de l'écran de bienvenu ainsi créé à «LAUNCHER».

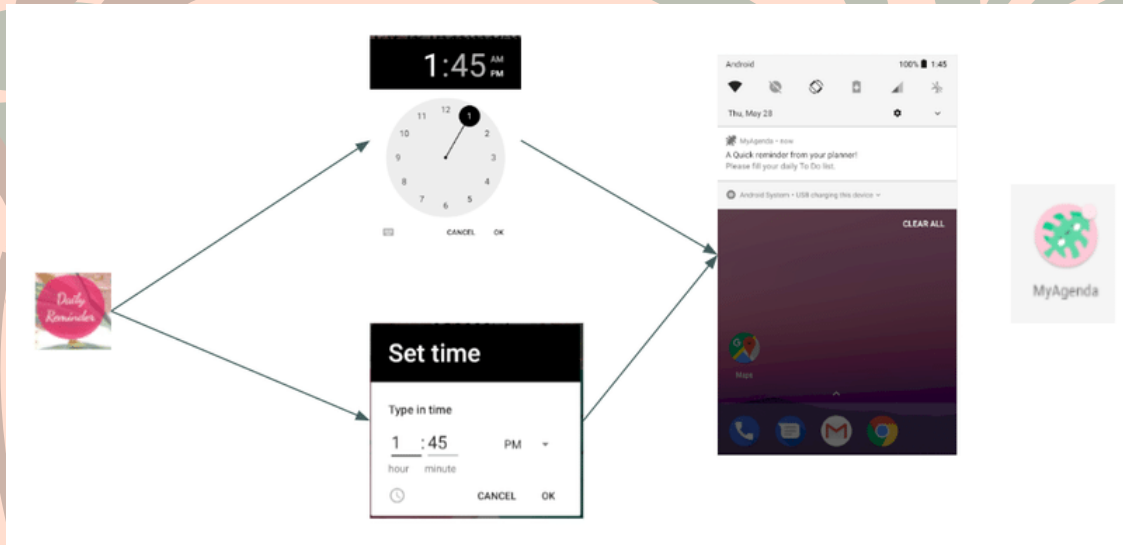


3.3. Ecran d'accueil :

L'écran d'accueil est l'activité définie par le couple de fichiers MainActivity.java dans le dossier java et activity_main.xml dans le dossier layout. Cet écran contient la date du jour affichée à l'aide de la classe Calendar et 4 boutons à identifiants uniques. Un bouton pour accéder à chaque compartiment de l'application et un bouton pour configurer un rappel journalier pour remplir la liste des tâches journalières. Dans la classe java, à l'aide des identifiants uniques des boutons, j'ai affecté des actions à mes boutons, pour que chaque bouton des 3 premiers dirige l'utilisateur vers le compartiment correspondant. Quant au dernier bouton le «Daily Reminder», il affiche l'horloge pour fixer l'heure journalier du rappel.



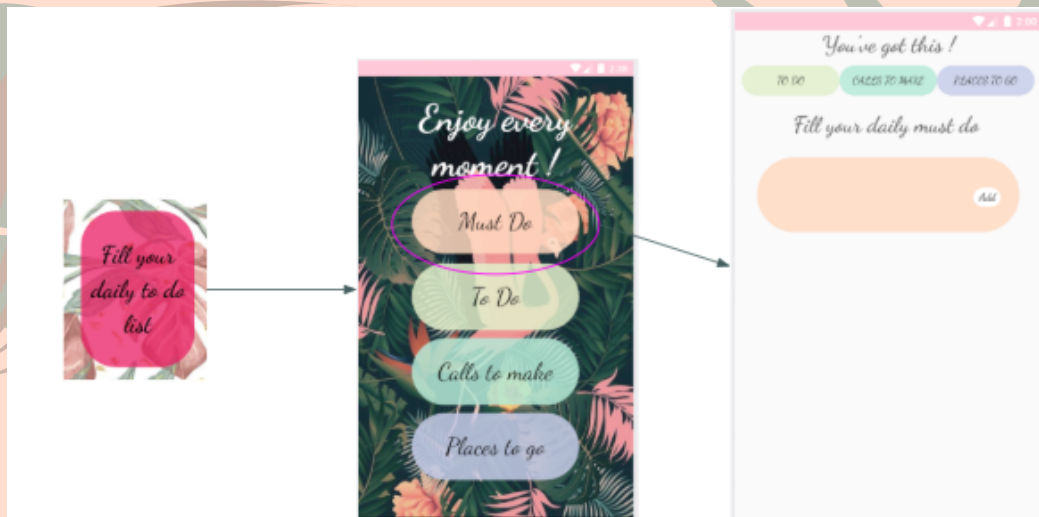
3.4. Rappel journalier et notification :



En cliquant sur le bouton «Daily Reminder» dans l'écran d'accueil, un «Time Picker Fragment» s'affiche pour permettre de choisir l'heure à laquelle l'application envoie une notification de rappel à son utilisateur pour remplir les tâches journalières à faire. Le «Time Picker Fragment» est mis en place avec la classe java `TimePickerFragement.java`. L'implémentation de l'alarme est effectuée par la fonction `onTimeSet` de la classe `MainActivity.java`. Cette fonction fait appel à la fonction `startAlarm` qui prend en paramètre l'heure précisée par l'utilisateur, démarre l'alarme à cette heure là et affiche une notification de rappel. Ce processus de création de notification se fait à l'aide des classes `AlertReceiver.java` et `NotificationHelper.java`. Cette dernière classe effectue le processus de création de la chaîne de notification dans le système du smartphone et personnalise le message à afficher.

3.5. Daily To Do List

Ce compartiment de l'application comporte les listes à remplir chaque jour. Il est défini par son activité principale `MainActivityDay.java` dans le dossier java et `activity_main_day` dans le dossier layout. La page d'accueil de ce compartiment comporte des boutons avec les noms des 4 listes. Le fichier java associé aux boutons des actions pour qu'en appuyant sur l'un des boutons, l'utilisateur se dirige vers la liste qu'il a choisie. Pour remplir toutes les listes et sans être obligé de revenir à l'écran d'accueil de ce compartiment, dans chaque liste il y a des boutons avec les noms des autres catégories de tâches en haut pour y accéder directement. Cela est expliqué par le schéma qui suit.



En cliquant sur le bouton «Fill you daily to do list» de l'écran d'accueil de l'application, l'utilisateur est dirigé vers la page d'accueil des tâches journalières qu'il doit accomplir. Chaque liste a un bouton avec dessus le nom de la liste. L'ajout et l'affichage des tâches dans chaque liste se fait avec les fonctions écrire (read) et lire (write) des classes java de type FileHelper et les données sont mis à jour dans un fichier d'extension *.dat. Dans l'application, l'ajout de la tâche se fait à l'aide du compartiment d'éditeur de texte (edittext) et le bouton «Add» qui met à jour la liste. La suppression d'une tâche est simplement effectué en cliquant dessus.

La liste «Must Do» est une activité définie par le couple de fichiers DailyMustDo.java et activity_daily_must_do.xml et elle est reliée aux classes FileHelper des jours de la semaine c-à-d FileHelperDay (Day=Monday, Tuesday, ... , Sunday) selon le jour que l'utilisateur remplit son «Daily To Do List» (Cela va être expliqué dans le compartiment des tâches de la semaine).

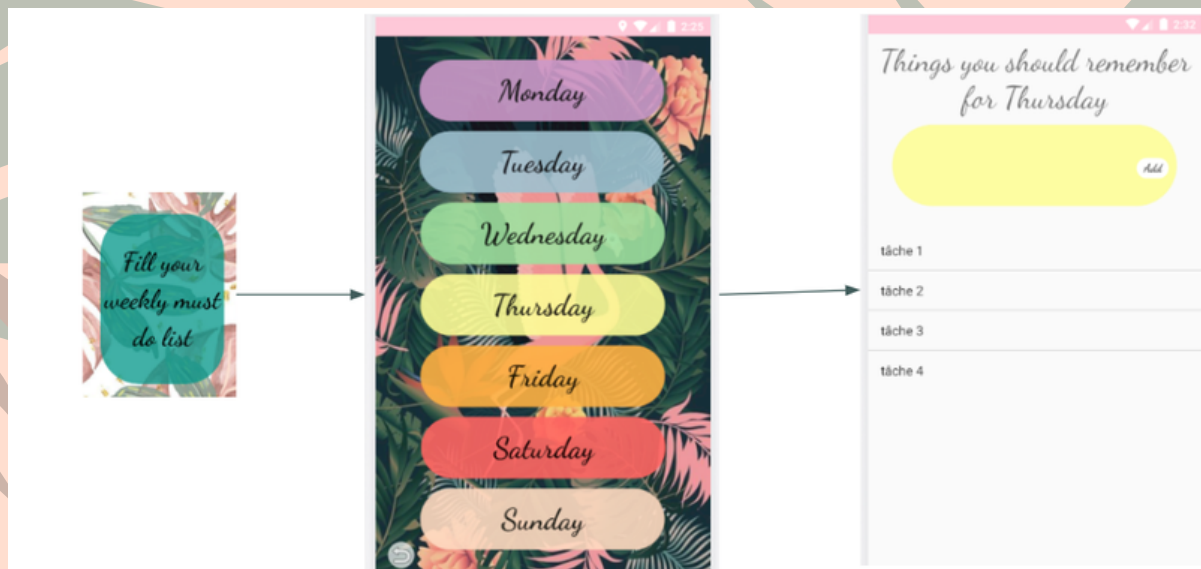
La liste «To Do» est une activité définie par le couple de fichiers DailyToDo.java dans le dossier java et activity_daily_to_do.xml dans le dossier layout et elle est relié à la classe java FileHelperToDo.

La liste «Calls To Make» est une activité définie par le couple de fichiers DailyCalls.java dans le dossier java et activity_daily_calls.xml dans le dossier layout et elle est relié à la classe java FileHelperCalls.

La liste «Places To Go» est une activité définie par le couple de fichiers DailyPlaces.java dans le dossier java et activity_daily_places.xml dans le dossier layout et elle est relié à la classe java FileHelperPlaces.

3.6. Weekly Must Do List

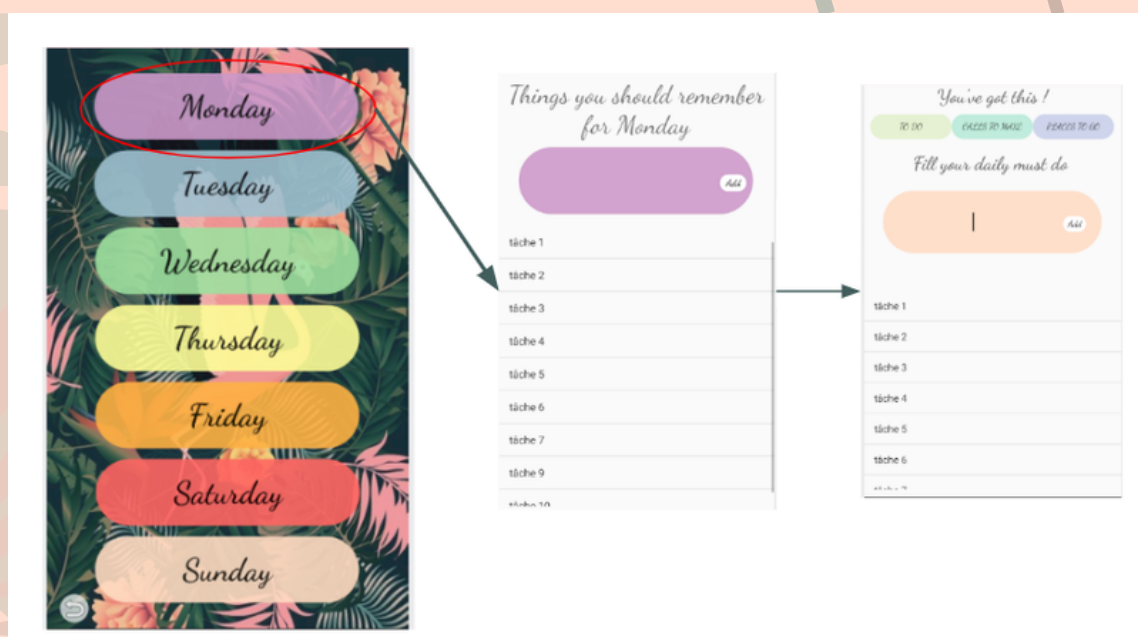
Ce compartiment de l'application comporte une liste pour chaque jour de la semaine (Monday -> Sunday). Il est défini par son activité principale WeeklyMustDo.java dans le dossier java et activity_weekly_must_do.xml dans le dossier layout. La page d'accueil de ce compartiment comporte 7 boutons sur lesquels est inscrit le nom du jour de la semaine et qui dirigent l'utilisateur vers la liste correspondante.



Le remplissage de ce compartiment peut se faire une seule fois en début de semaine pour permettre à l'utilisateur d'avoir une vue globale de sa semaine et des tâches qu'il ne doit pas oublier. Ce compartiment est aussi modifiable tout au long de la semaine pour ajout ou suppression. Le point fort de l'application c'est que pour chaque jour de la semaine, les tâches correspondantes à ce jour sont directement transférés vers la liste journalière des tâches que l'utilisateur doit faire et ceci évitera la consultation des tâches du jour dans le compartiment des «Weekly Must Do».

L'exemple dans la figure qui suit montre qu'en écrivant un nombre de tâches (illimités en nombre et en taille du contenu) dans la liste de Lundi (vu que j'écris le rapport un lundi), ces mêmes tâches sont écrites dans la liste des «Must Do» ou les tâches que l'utilisateur doit faire. Toute modification dans l'une des listes affectera les deux en même temps (ajout ou suppression).

La liste «Day» est une activité définie par le couple de fichiers Day.java dans le dossier java et activity_day.xml dans le dossier layout et elle est reliée à la classe java FileHelperDay. Avec Day = Monday, Tuesday, ..., Sunday.



3.7. Grocery Shopping List

Cette partie de l'application est une liste des achats pour l'utilisateur. Elle contient une liste de ses besoins qu'il a ajouté au fur et à mesure d'une période. Pour faciliter le remplissage de la liste des besoins, ce compartiment est réparti en listes des rayons que l'utilisateur trouve généralement dans un magasin. La liste des produits à acheter vise à cibler les achats vers les produits que l'utilisateur en a réellement besoin. Ce compartiment est défini par son activité principale Grocery.java dans le dossier java et activity_grocery.xml dans le dossier layout. La page d'accueil de ce compartiment contient des boutons images avec des images des produits du rayon sur le bouton, ces boutons images dirigent vers la liste correspondante. Dans chaque liste, les autres boutons images des autres rayons se trouvent en haut pour faciliter le passage d'une liste rayon à une autre sans revenir à la page d'accueil.



Le remplissage de ces listes se fait avec le même principe de remplissage de toutes les autres listes de l'application, c-à-d un FileHelper à toute liste.

La liste «Rayon» est définie par l'activité Fruits.java et activity_rayon.xml et elle est reliée au FileHelperRayon.java avec Rayon = Fruits, Vegetables, Dairy, FishMeat, Pasta, Snacks, Paper, Toilet et More.