

# Interactions avec l'utilisateur

## Approche MVC

---

Laboratoire n°3

DAA 23-24

# Développement Application Android

---

Département TIC

**Auteurs:**

Anthon Pirakas, Chollet Bastian, Cochet Yvan

**Enseignant:**

Dutoit Fabien

**Assistant :**

Goujgali Ilias

**Classe:**

B

**Salle de classe :**

C23

# **Table des matières**

---

1. [Introduction](#)
3. [Implementation](#)
4. [Questions](#)
5. [Conclusion](#)

# Introduction

---

Dans le cadre du cours de DAA, les étudiants doivent effectuer plusieurs manipulations destinées à appréhender la mise en place des différents composants Android de base pour la réalisation d'une Activité proposant un formulaire permettant d'éditer les informations d'une personne. Ce document sert à répondre aux questions du document d'instruction.

## Implementation

---

Comme dans le premier laboratoire, nous avons utilisé le linkage automatique des vues en modifiant le fichier gradle. Comme demandé, nous avons uniquement utilisé un `ConstraintLayout`, sans ajout de layout supplémentaire. Comme nous avons deux sous-classes, il nous suffit d'afficher les composants des sous classes selon la sélection de l'utilisateur.

Dans la méthode `onCreate()`, nous avons créé la fonction `fillForm` qui permet de remplir automatiquement tous nos champs de formulaire. Comme demandé au lancement de notre application, le formulaire est vide. Mais si on décommente la ligne dans notre code alors cette fonction va créer l'`exampleWorker`. On peut bien entendu créer un objet `student` si on lui passe à la fonction comme argument.

Nous avons choisi d'utiliser `MaterialDatePicker` pour le choix de la date, car il est plus moderne. Nous avons limité le range de choix à l'aide de `CalendarConstraints` issue de la même librairie. Pour mettre à zéro notre formulaire, nous passons sur nos widgets en effectuant du `patern matching` et les settons à la valeur souhaitée.

Lors de la création de `Person`, nous avons décidé de faire de la validation d'input très basique. Si l'utilisateur ne rentre pas de données dans un champ du formulaire et confirme alors `createPerson` va retourner `NULL`. Nous effectuons cela pour éviter les crashes inopinés de notre application. Nous ne vérifions pas la cohérence des données. Pour la partie spinner, nous avons géré de la même manière, si l'utilisateur sélectionne l'option "Sélectionner" et envoie le formulaire, alors le retour de la fonction sera à `NULL`.

## Questions

---

**1. Pour le champ `remark`, destiné à accueillir un texte pouvant être plus long qu'une seule ligne, quelle configuration particulière faut-il faire dans le fichier XML pour que son comportement soit correct ? Nous pensons notamment à la possibilité de faire des retours à la ligne, d'activer le correcteur orthographique et de permettre au champ de prendre la taille nécessaire.**

Voici la balise `EditText` dans laquelle on peut voir l'attribut `android:inputType="textMultiLine"` ce qui va nous permettre d'ajouter plusieurs lignes et donc des retours à la ligne comme demandé. Pour le correcteur orthographique, il est activé par défaut donc il n'y a pas besoin d'intervention de notre part. La taille du champs va s'adapter à la taille de notre texte donc il va évoluer selon ce que l'on tape et il va prendre la taille nécessaire.

```
<EditText
    android:id="@+id/additional_remarks_input"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:inputType="textMultiLine"
    android:autofillHints="name"
    app:layout_constraintTop_toBottomOf="@+id/additional_remarks_title"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />
```

**2. Pour afficher la date sélectionnée via le DatePicker nous pouvons utiliser un DateFormat permettant par exemple d'afficher 12 juin 1996 à partir d'une instance de Date. Le formatage des dates peut être relativement différent en fonction des langues, la traduction des mois par exemple, mais également des habitudes régionales différentes : la même date en anglais britannique serait 12th June 1996 et en anglais américain June 12, 1996. Comment peut-on gérer cela au mieux ?**

Pour gérer ça, nous avons utilisé la fonction `DateFormat.getDateInstance(DateFormat.LONG, Locale.getDefault())`. L'option "DateFormat.LONG" permet d'afficher le texte en entier et l'option "Locale.getDefault()" permet de prendre le format locale du système sur lequel le programme est en cours d'exécution.

**3 .Veuillez choisir une question en fonction de votre choix d'implémentation :**

**a. Si vous avez utilisé le DatePickerDialog1 du SDK. En cas de rotation de l'écran du smartphone lorsque le dialogue est ouvert, une exception `android.view.WindowLeaked` sera présente dans les logs, à quoi est-elle due ?**

**b. Si vous avez utilisé le MaterialDatePicker2 de la librairie Material. Est-il possible de limiter les dates sélectionnables dans le dialogue, en particulier pour une date de naissance il est peu probable d'avoir une personne née il y a plus de 110 ans ou à une date dans le futur. Comment pouvons-nous mettre cela en place ?**

Nous avons utilisé MaterialDatePicker, on doit définir des contraintes pour limiter les dates sélectionnables. Nous avons utilisé l'API `CalendarConstraints` pour spécifier une plage de dates autorisées. La date de début sera -110 ans (1913) depuis aujourd'hui et la date de fin sera le mois actuelle.

**4.1 Lors du remplissage des champs textuels, vous pouvez constater que le bouton « suivant » présent sur le clavier virtuel permet de sauter automatiquement au prochain champ à saisir, cf. Fig. 2. Est-ce possible de spécifier son propre ordre de remplissage du questionnaire ?**

Oui, nous pouvons modifier de sauter au champ que l'on désire en ajoutant l'attribut `android:imeOptions="actionNext"` et l'attribut `android:nextFocusForward` en lui spécifiant la clé vers le prochain composant que l'on souhaite accéder.

**4.2 Arrivé sur le dernier champ, est-il possible de faire en sorte que ce bouton soit lié au bouton de validation du questionnaire ? Hint : Le champ remark, multilignes, peut provoquer des effets de bords en fonction du clavier virtuel utilisé sur votre smartphone. Vous pouvez l'échanger avec le champ e-mail pour faciliter vos recherches concernant la réponse à cette question.**

Pour faire cela, il faut pour cela utiliser l'attribut `android:imeOptions` du dernier champ, et lui donner la valeur "actionDone". On va ensuite créer un `setOnEditorActionListener` dans le `onCreate` pour effectuer une simulation du bouton OK pour valider le formulaire.

**5 Pour les deux Spinners (nationalité et secteur d'activité), comment peut-on faire en sorte que le premier choix corresponde au choix null, affichant par exemple « Sélectionner » ? Comment peut-on gérer cette valeur pour ne pas qu'elle soit confondue avec une réponse ?**

Nous avons décidé de changer les valeurs dans la première valeurs des items dans `string.xml` pour qu'il affiche la valeur "Sélectionner". Nous avons uniquement décidé de gérer cette partie avec la validation d'input. Si l'utilisateur décide de choisir la valeur "Sélectionner" et clique sur OK alors la fonction `createPerson` va simplement retourner NULL donc l'utilisateur ne pourra pas envoyer le formulaire avec la valeur "Sélectionner".

## Conclusion

---

Ce troisième laboratoire de DAA, nous a permis de faire des manipulations des composants Android de base. La mise en place d'une activité avec un formulaire pour éditer les informations d'une personne a permis de concrétiser les concepts enseignés.