Développement Android
Laboratoire n°3

Interactions avec l'utilisateur
Approche MVC

01.11.2023

Introduction

Ce laboratoire est constitué de plusieurs manipulations et questions théoriques 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.

Manipulations

1. Mise en place

Pour ce laboratoire, vous allez partir d'une application par défaut « empty view activity » à laquelle vous intégrerez les ressources que nous vous fournissons, à savoir :

- Le fichier *Person.kt* qui contient la classe abstraite *Person* ainsi que ses deux sous-classes *Worker* et *Student*. Nous vous avons également mis à disposition deux instances statiques exemples : *exampleWorker* et *exampleStudent* contenant des données illustratives. Le diagramme *UML* correspondant est disponible dans les slides de présentation ;
- Le fichier *string.xml* contenant la définition de textes et de tableaux de valeurs utilisables dans ce laboratoire. Il s'agit d'une proposition pour vous faire gagner du temps, sentez-vous libre de l'adapter ;
- L'image vectorielle *cake.xml* représentant un gâteau d'anniversaire.

2. Développement de la vue

Nous vous demandons à présent de réaliser l'interface graphique de notre application. Il s'agit de concevoir le fichier *layout* intégrant et mettant en page tous les textes, les widgets et les différents éléments de notre interface graphique. Vous trouverez sur la Fig. 1 une proposition de mise en page, nous vous demandons de réaliser votre UI à l'aide d'un *ConstraintLayout*. Vous veillerez à ne pas

imbriquer un trop grand nombre de *layouts* pour ne pas péjorer les performances, un bonus vous sera accordé si vous arrivez à réaliser votre interface graphique avec un seul *layout*.

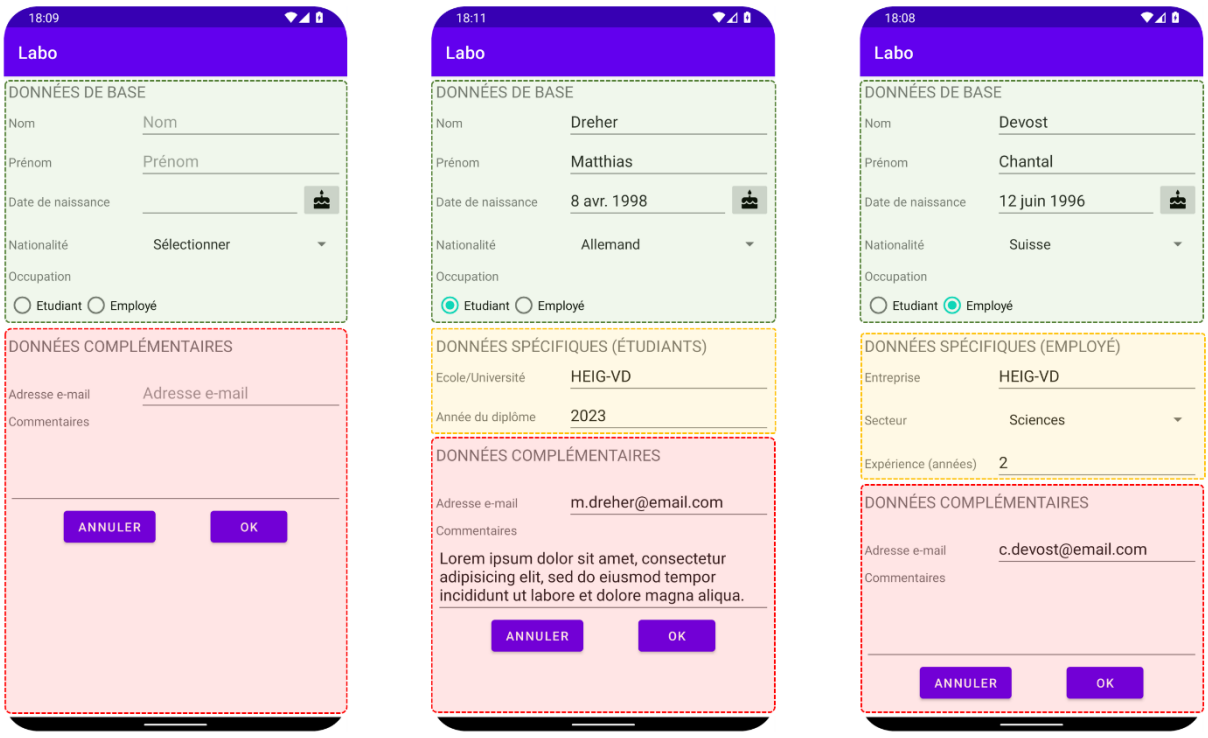

Figure 1 - Exemple de l'interface graphique permettant d'éditer une Personne. Les zones en couleur servent à mettre en évidence les différentes parties du formulaire (en vert : données de base d'une personne, en orange : particularités d'un employé ou d'un étudiant, en rouge les champs complémentaires en commun).

Votre interface devra permettre d'éditer des *Person*, soit des *Worker* ou des *Student*, en mettant à disposition les widgets adaptés à la saisie des différents champs de ces classes, en particulier :

- Des *EditText*, ou similaire, pour la saisie des informations textuelles ou numériques ;
- Des *Spinner* pour le choix d'une valeur parmi une liste (nationalité et secteur d'activité d'un employé) ;
- Des *RadioButton* pour le choix du type de la *Person*, un *Worker* ou un *Student* ;
- La saisie de la date d'anniversaire se fera à l'aide d'un *DatePicker*, un dialogue qui s'ouvrira lors de l'appui sur un bouton à côté du champ de saisie ou sur le champ de saisie (qui ne doit pas pouvoir être directement édité par l'utilisateur à l'aide du clavier). Si la date d'anniversaire n'est pas encore définie le *DatePicker* sera présélectionné avec la date du jour, si elle est définie le *DatePicker* sera présélectionné avec celle-ci.

Votre interface devra s'adapter à la saisie d'un type de *Person* ou de l'autre en fonction du choix de l'utilisateur, les zones de saisie dédiées à des informations concernant uniquement un étudiant ne devront pas être affichées lors de l'édition d'un employé, et vice-versa. Également vous mettrez deux boutons à disposition permettant de : 1) vider le formulaire, 2) valider le formulaire.

3. Le contrôleur

Il s'agit de l'activité qui accueillera la vue et implémentera les différentes fonctionnalités devant être offertes :

- L'initialisation de la vue en fonction d'un objet existant du modèle, par exemple *exampleWorker* ou *exampleStudent* ;
- De créer une instance d'un *Student* ou d'un *Worker* lors de l'appui sur le bouton de validation du formulaire avec les données saisies dans celui-ci et l'affichage de l'instance créée dans les logs (méthode *toString()* de la *class*) ;
- Supprimer le contenu de tous les éléments de la GUI lors de l'appui sur le bouton correspondant.

4. Questions complémentaires

Veuillez répondre aux 5 questions suivantes. Pour chacune d'entre elles, vous développerez votre réponse et l'illustrerez par des extraits de code.

4.1 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.

4.2 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 ?

4.3 Veuillez choisir une question en fonction de votre choix d'implémentation :

a. Si vous avez utilisé le *DatePickerDialog*[1] 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 *MaterialDatePicker*[2] 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 ?

4.4 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 ?
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.

[1] https://developer.android.com/reference/android/app/DatePickerDialog
[2] https://github.com/material-components/material-components-android/blob/master/docs/components/DatePicker.md

4.5 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 ?

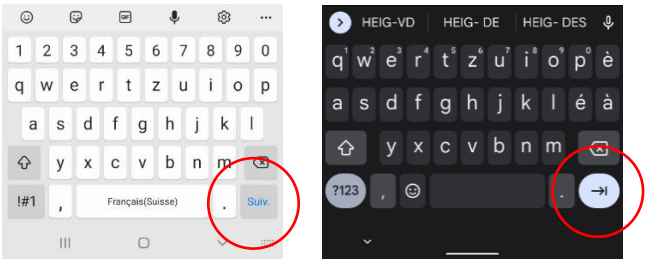

Figure 2 - Claviers virtuels (Samsung à gauche, et Google à droite) lors du remplissage de champs textuels. Les cercles rouges identifient un bouton permettant de passer automatiquement au prochain champ sans devoir fermer le clavier.

Durée / Evaluation

- 4 périodes, à rendre le mardi **14.11.2023** à **23h55** au plus tard.
- Pour le rendu de votre code, nous vous demandons de bien vouloir zipper votre projet Android Studio en veillant à bien supprimer les dossiers *build* (à la racine et dans *app/*) pour limiter la taille du rendu. Vous remettrez également un document **pdf** comportant les explications sur l'implémentation de votre solution ainsi que les réponses aux questions posées.
- Merci de rendre votre travail sur *CyberLearn* dans un zip unique. N'oubliez pas d'indiquer vos noms dans le code, sur vos réponses et de commenter vos solutions.