Programmation mobile

Compte-rendu TP3

Fragment, réseau, services, fichiers, ...

Louis Jean
Master 1 IMAGINE
Université de Montpellier
N° étudiant : 21914083
https://github.com/louis-jean0/HAI811I-mobile

7 avril 2024

Table des matières

1 Introduction

Ce travail pratique a été conçu pour visiter plusieurs aspects clés du développement sous Android. À travers une série d'exercices progressifs, nous avons eu l'opportunité d'explorer la création d'interfaces utilisateur, la gestion des fragments, la persistance des données, l'intégration de services réseau, et le développement de services en arrière-plan.

2 Fragment

2.1 Objectif

L'objectif de cet exercice était de créer une application Android utilisant des fragments pour permettre à l'utilisateur de s'inscrire. L'inscription impliquait la saisie de diverses informations personnelles et de préférences, suivie d'une étape de validation où l'utilisateur pouvait vérifier et confirmer ces informations.

2.2 Mise en œuvre

2.2.1 Fragment de saisie

Le premier fragment, `FragmentSaisie`, a été conçu pour collecter les informations de l'utilisateur. Il contenait des champs *EditText* pour le nom, le prénom, l'adresse email, et le numéro de téléphone, ainsi qu'un *DatePickerDialog* pour la date de naissance et plusieurs *CheckBox* pour les centres d'intérêt.
Pour améliorer l'expérience utilisateur, un *DatePickerDialog* a été implémenté permettant une sélection intuitive de la date de naissance. Les données saisies étaient récupérées et stockées à l'aide d'une méthode dédiée, `passerLesDonnees`, qui était appelée lorsque l'utilisateur cliquait sur le bouton "Soumettre".

2.2.2 Fragment d'affichage

Le second fragment, `FragmentAffichage`, avait pour but d'afficher un résumé des informations saisies pour confirmation. Les données étaient transmises du `FragmentSaisie` au `FragmentAffichage` via un *Bundle*, exploitant une interface de callback pour la communication entre fragments.

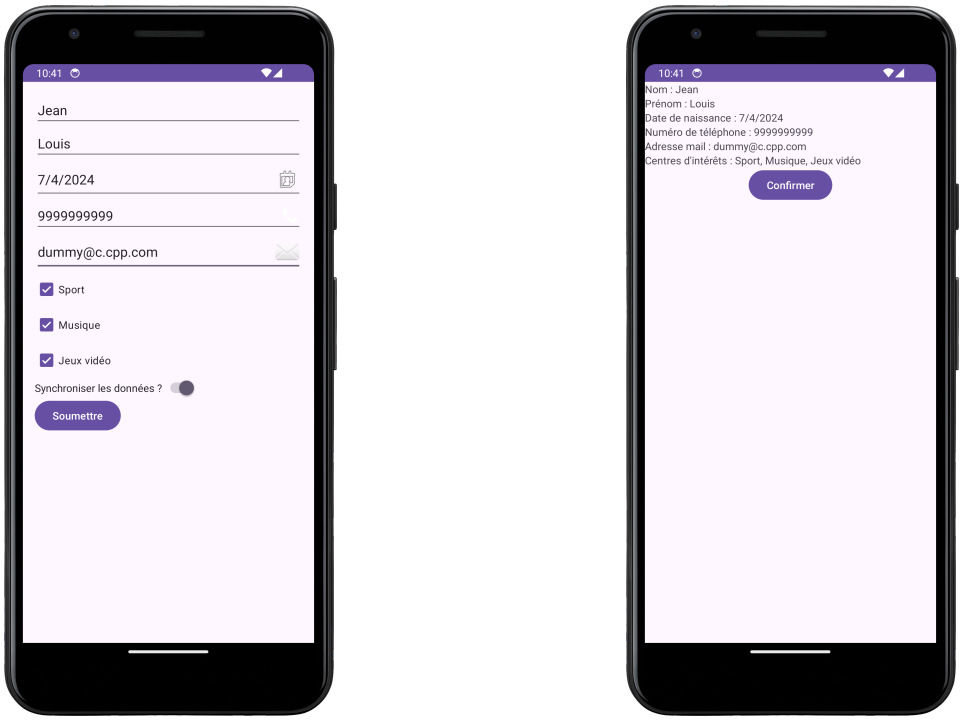
2.2.3 Affichage et mise en page

Une attention particulière a été accordée à la mise en page des fragments pour assurer une présentation claire et accessible des informations. Le `fragment_saisie.xml` utilisait principalement des *LinearLayouts* pour organiser les champs de saisie, tandis que le `fragment_affichage.xml` était structuré pour afficher les informations de manière séquentielle, améliorant ainsi la lisibilité.

2.3 Défis rencontrés et solutions apportées

Un défi rencontré a été la gestion des événements de clic sur l'EditText de la date de naissance, qui devait ouvrir un *DatePickerDialog*. Ce problème a été résolu en désactivant le focus de l'EditText pour éviter l'ouverture du clavier virtuel et en attachant correctement un gestionnaire d'événements de clic.

2.4 Résultats



(a) Fragment de saisie (b) Fragment d'affichage

Figure 1: Fragments

3 Fichier et persistance de l'état d'une activité

3.1 Objectif

L'objectif de cet exercice était d'étendre l'application développée précédemment en y intégrant des fonctionnalités de persistance de données. Cela impliquait le stockage des données saisies par l'utilisateur dans un fichier lors de la validation de l'inscription, ainsi que la gestion de la persistance de certaines de ces données lors de la navigation entre les fragments de l'application.

3.2 Mise en œuvre

3.2.1 Stockage des données

Les données saisies par l'utilisateur dans le fragment de saisie sont stockées dans un fichier texte interne à l'application lorsqu'il valide son inscription. Ces données sont structurées au format JSON pour faciliter leur manipulation et leur lecture ultérieures.

- Un objet `JSONObject` a été utilisé pour regrouper les différentes informations de l'utilisateur (nom, prénom, date de naissance, etc.) ainsi que ses centres d'intérêt, ces derniers étant gérés comme un tableau JSON.
- Les données sont ensuite sérialisées en chaîne de caractères JSON et écrites dans un fichier nommé `userData.json`, situé dans le dossier `data` du package de l'application (il faut fouiller dans Android Studio), grâce à la classe `FileOutputStream`.

3.2.2 Navigation et persistance des données

Un bouton "Retour" a été ajouté au fragment d'affichage, permettant à l'utilisateur de revenir au fragment de saisie. Pour améliorer l'expérience utilisateur, les données déjà saisies sont rendues persistantes et réaffichées dans le formulaire de saisie, même après la navigation de l'utilisateur.

- Cette persistance a été réalisée en stockant temporairement les données dans un `Bundle` associé à l'activité principale (`MainActivity.userDataBundle`).
- Lorsque l'utilisateur retourne au fragment de saisie, les données stockées sont récupérées et les champs correspondants sont pré-remplis avec ces valeurs.

3.3 Défis rencontrés et solutions apportées

L'un des principaux défis a été la gestion efficace de la navigation et de la persistance des données entre les fragments. La solution retenue a été d'utiliser un `Bundle` statique dans l'activité principale comme mécanisme de stockage temporaire.

3.4 Résultats

```
{"nom":"Jean","prenom":"Louis","naissance":"7\/4\/2024","telephone":"9999999999","mail":"dummy@c.cpp.com","centresInterets":["Sport","Musique","Jeux vidéo"]}
```

Figure 2: Fichier userData.json généré

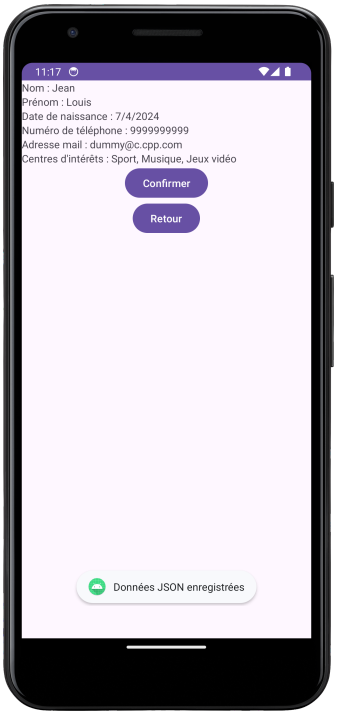


Figure 3: Enregistrement du fichier JSON

4 Réseau et utilisation de services web

4.1 Objectif

L'exercice 3 visait à enrichir l'application développée précédemment avec une fonctionnalité de réseau permettant de télécharger des données utilisateur depuis un service web REST. L'ajout d'un bouton "Télécharger" à l'interface utilisateur devait permettre de lancer cette opération. Ensuite, l'application devait parser un fichier JSON contenant les données de l'utilisateur et récupérer ses informations.

4.2 Approche conceptuelle

Bien que l'exercice n'ait pas été mis en œuvre faute de disponibilité d'une API web personnelle et de contraintes de temps, l'approche aurait consisté à :

- Ajouter un bouton "Télécharger" dans le fragment de saisie pour initier le téléchargement des données.
- Utiliser les API Android telles que 'HttpURLConnection' pour réaliser une requête GET vers un service web REST.
- Parser le fichier JSON récupéré en utilisant 'JSONObject' pour extraire les données de l'utilisateur.

Cette approche aurait permis de simuler une interaction réelle avec un service web et de pratiquer le parsing de données structurées.

4.3 Limitations

L'absence d'une API web personnelle a empêché la réalisation pratique de cet exercice, limitant l'expérience à la conceptualisation des étapes nécessaires sans implémentation réelle.

5 Service

5.1 Objectif

L'exercice 4 avait pour but de définir un service au sein de l'application qui prendrait en charge le téléchargement et le parsing des données utilisateur, potentiellement sans nécessiter une interaction directe avec l'interface utilisateur.

5.2 Approche conceptuelle

L'implémentation envisagée pour cet exercice aurait impliqué :

- La création d'un service Android, potentiellement un 'IntentService', pour gérer les opérations de téléchargement en arrière-plan.
- L'utilisation des mêmes mécanismes de requête réseau et de parsing JSON que ceux identifiés dans l'exercice 3 pour traiter les données de l'utilisateur.
- Le déclenchement du service depuis l'interface utilisateur ou au démarrage de l'application.

5.3 Limitations

Comme pour l'exercice 3, l'absence de ressources personnelles pour héberger une API web a limité l'application pratique de cet exercice à la planification théorique.

6 Conclusion

Le travail pratique proposé a constitué une exploration approfondie des différentes facettes du développement d'applications Android, allant de la manipulation de l'interface utilisateur à l'intégration de fonctionnalités réseau complexes. Bien que tous les exercices n'aient pas été réalisés en pratique, principalement en raison de contraintes externes telles que l'absence d'une API web personnelle et de limitations de temps, la planification et la conceptualisation de ces tâches ont néanmoins permis de souligner l'importance de comprendre et d'appliquer les principes de développement Android dans des scénarios réels.

Merci pour le temps et l'attention que vous avez consacrés à la lecture de ce compte-rendu.