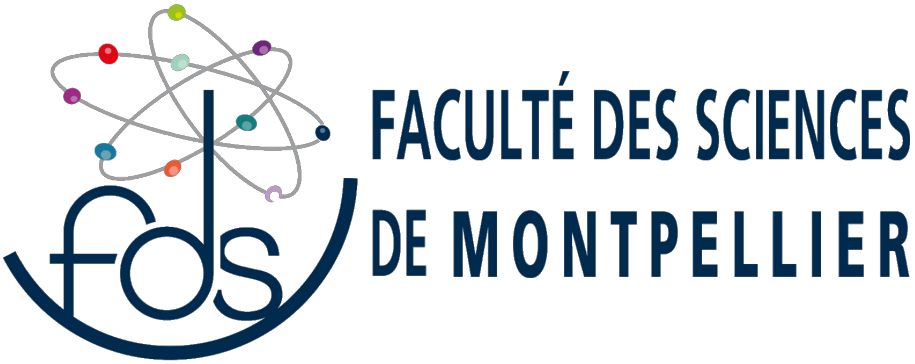

Rapport de TP1 Programmation Mobile

Bases du développement Android

Master GL

Faculté des Sciences

Auteur : Léo Ponchon

Date : 15 février 2026

Lien dépôt : Depot TP1

Table des matières

Résumé du projet

Ce TP1 avait pour objectif de découvrir et maîtriser les bases du développement Android en Java natif avec Android Studio. L'ensemble des neuf exercices demandés a été réalisé. Chaque exercice a été placé dans une activité distincte afin de garder une architecture claire et facile à comprendre.

Le projet couvre la création d'interfaces graphiques, la gestion des événements utilisateurs, l'internationalisation, la navigation entre activités ainsi que l'utilisation des intents. Certaines parties ont été développées au-delà de ce qui était strictement demandé, notamment pour les exercices 8 et 9 qui ont été rendus entièrement fonctionnels.

Étape 1 — Découverte du projet Android

Exercice 1 : Structure d'un projet

Le premier exercice consistait à analyser la structure d'un projet Android. Cela a permis de comprendre le rôle du fichier `AndroidManifest.xml`, l'organisation du dossier `res/` contenant les layouts et ressources, ainsi que le dossier `java/` qui regroupe les activités.

Cette étape était importante pour comprendre comment les différents éléments d'une application Android interagissent entre eux.

Exercice 2 : Hello World et inspection dynamique

Au lieu de simplement afficher un message statique, l'activité `Exo2Activity` a été développée pour inspecter dynamiquement l'application. Elle affiche différentes informations sur les composants déclarés, ce qui permet de mieux comprendre le fonctionnement interne du projet.

Cette approche va un peu plus loin qu'un simple "Hello World" classique.

Étape 2 — Interfaces et gestion des événements

Exercice 3 : Formulaire en XML et en Java

Cet exercice a été réalisé en deux versions. La première utilise un layout XML classique pour définir l'interface. La seconde version construit entièrement l'interface en Java, sans fichier XML.

Un problème de crash est apparu dans la version Java au début, mais il a été identifié puis corrigé. Cela a permis de mieux comprendre la gestion du contexte et la création dynamique de vues.

Exercice 4 : Internationalisation

L'application prend en charge deux langues, le français et l'anglais. Les textes sont définis dans les fichiers `strings.xml`. Lorsque la langue du téléphone change, l'application adapte automatiquement son affichage.

Cette mise en place respecte les bonnes pratiques recommandées sous Android.

Exercice 5 : Gestion des événements

Lors de la validation d'un formulaire, une `AlertDialog` est affichée pour demander confirmation à l'utilisateur. Cela permet de gérer correctement les interactions.

Durant le développement, une zone d'affichage des logs avait été ajoutée pour faciliter le débogage. Elle a été supprimée dans la version finale afin de garder une application plus propre. Les instructions de log ont également été retirées.

Étape 3 — Navigation et communication entre activités

Exercice 6 : Intent explicite

La navigation entre plusieurs activités est réalisée à l'aide d'intents explicites. Les données sont transmises grâce à la méthode `putExtra()` et récupérées dans les activités suivantes.

Cette partie permet de comprendre comment structurer une application composée de plusieurs écrans.

Exercice 7 : Intent implicite

Un appel téléphonique est déclenché à l'aide d'un intent implicite utilisant `ACTION_DIAL`. Ce choix est volontaire car il ne nécessite pas de permission sensible.

Un bug empêchait initialement le lancement de l'appel. Une vérification avec `resolveActivity()` indiquait qu'aucune application téléphonique n'était disponible, même sur un appareil physique. Le problème venait des restrictions récentes d'Android sur la visibilité des applications. La solution a été de supprimer cette vérification et de lancer directement l'intent, ce qui a corrigé le comportement.

Étape 4 — Approfondissements

Exercice 8 : Horaires de trains

Même si l'énoncé demandait uniquement une interface graphique, une version fonctionnelle a été développée. L'application effectue un appel réseau asynchrone vers l'API de la SNCF, récupère une réponse au format JSON puis affiche dynamiquement les horaires obtenus.

Cela a permis de manipuler les appels HTTP et le traitement de données distantes.

Exercice 9 : Agenda

Un agenda simple mais fonctionnel a été implémenté. Il permet d'ajouter un événement, de l'afficher selon la date choisie et de le supprimer. Les données sont conservées en mémoire pendant l'exécution de l'application.

Cet exercice a permis de travailler la gestion dynamique des listes et des interactions utilisateur.

Conclusion

Ce TP1 a permis d'acquérir les bases essentielles du développement Android. Les différents exercices ont permis de comprendre la structure d'un projet, la création d'interfaces, la gestion des événements ainsi que la navigation entre activités, c'est une très bonne entrée en matière pour attaquer le TP des capteurs et le futur projet.

L'application finale est fonctionnelle, structurée et cohérente. Certaines parties ont été approfondies afin d'aller un peu plus loin que les exigences minimales, ce qui rend le projet plus complet et plus abouti.

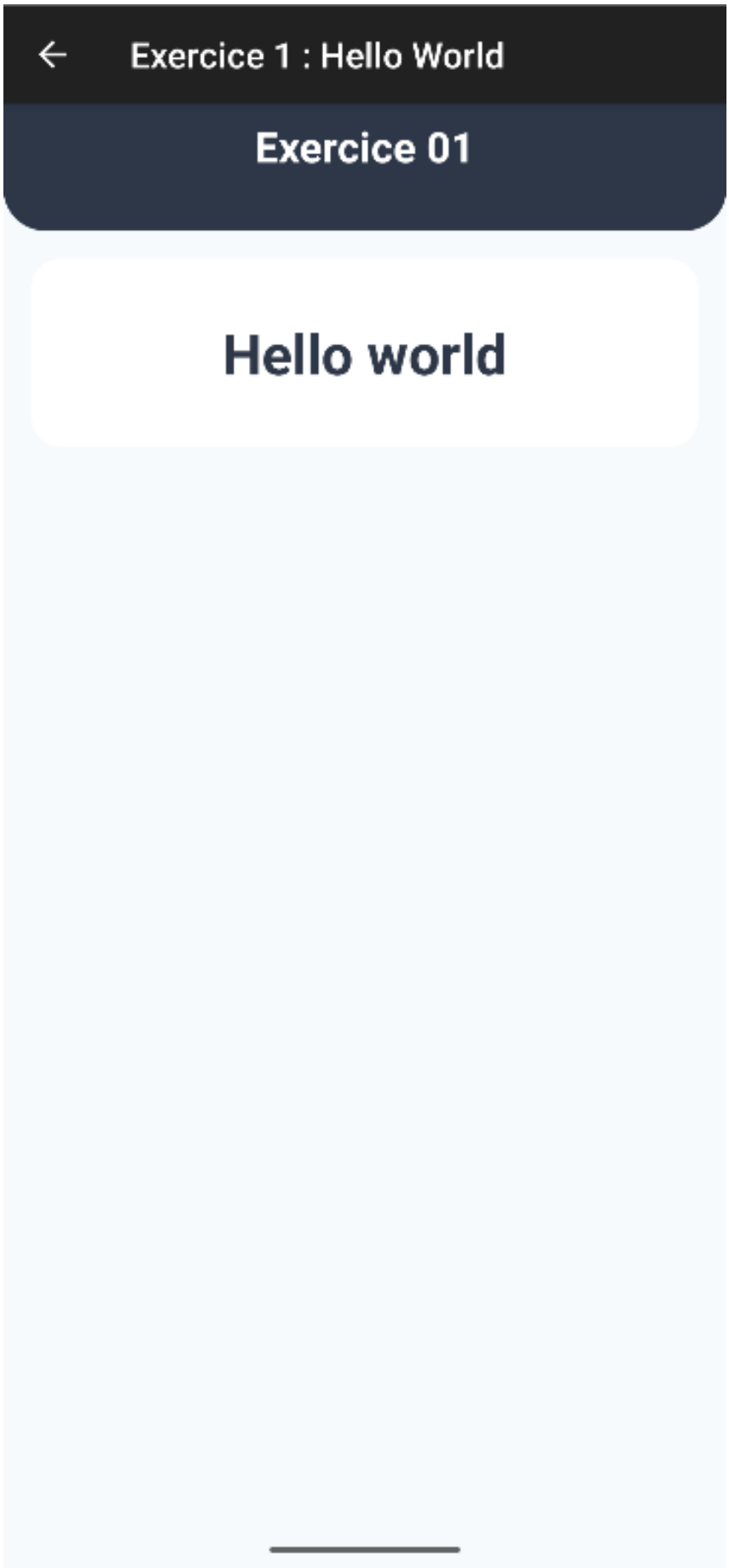

FIGURE 1 – Structure du projet Android

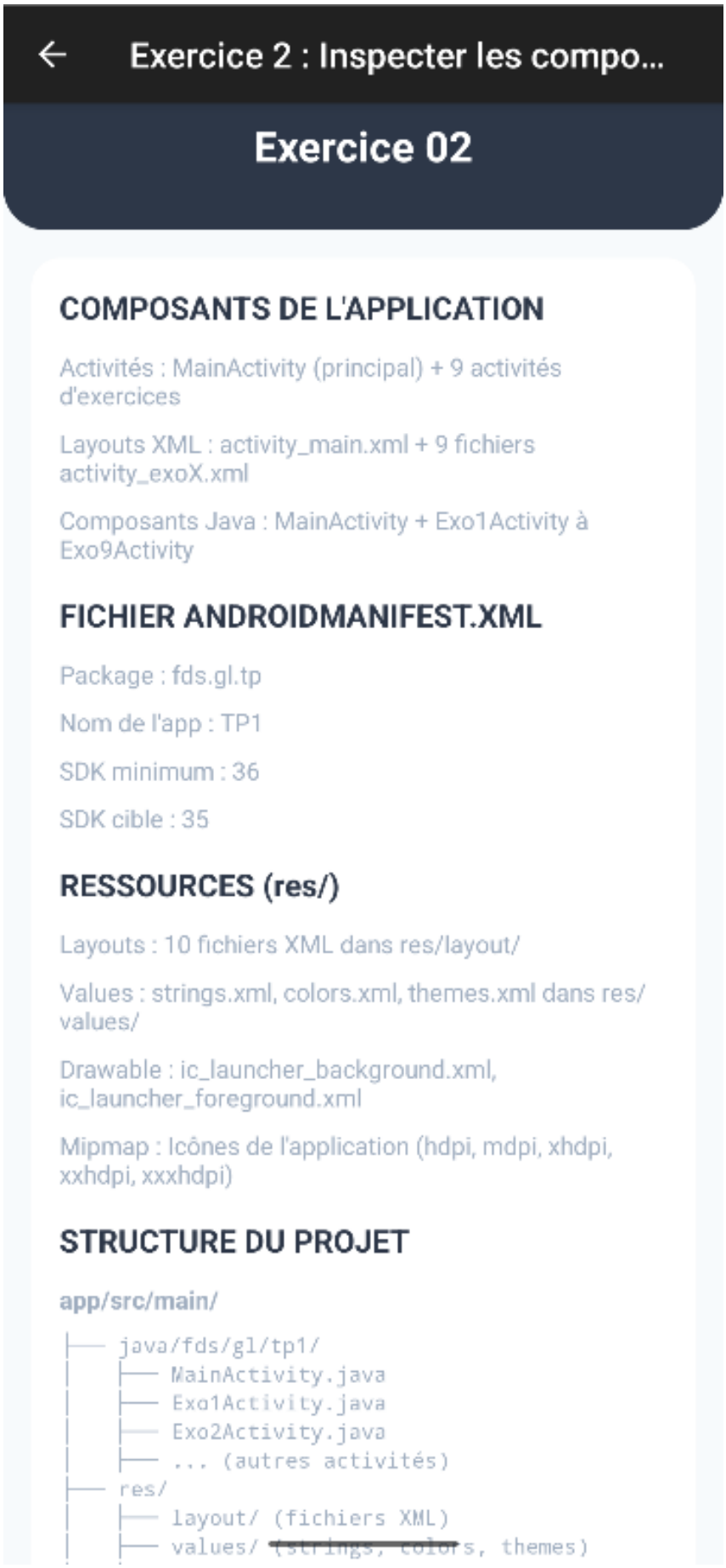

FIGURE 2 – Exercice 2 — Inspection dynamique de l'application

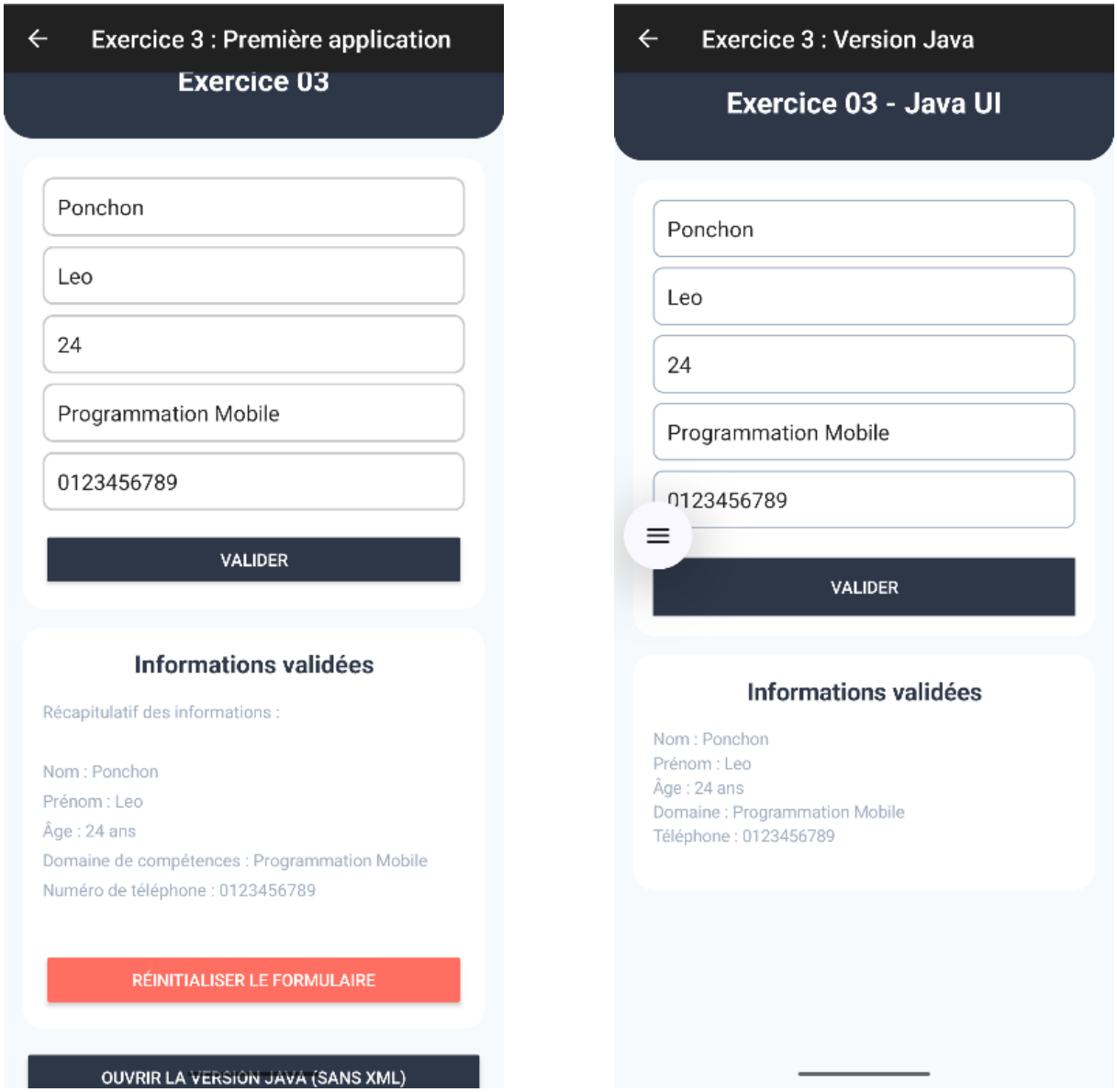

FIGURE 3 – Exercice 3 — Formulaire en XML (gauche) et en Java (droite)

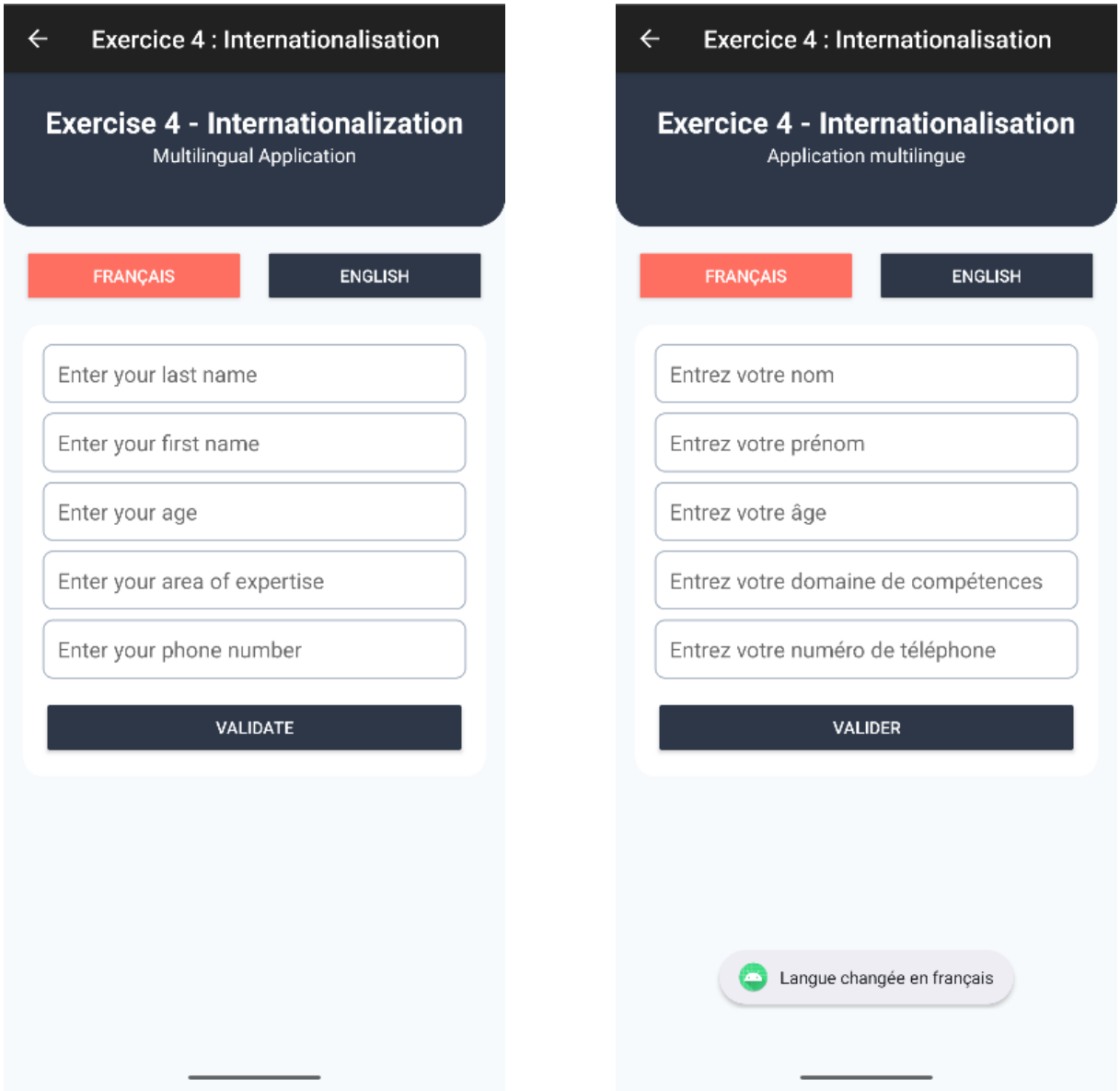

FIGURE 4 – Exercice 4 — Internationalisation : français (gauche) et anglais (droite)

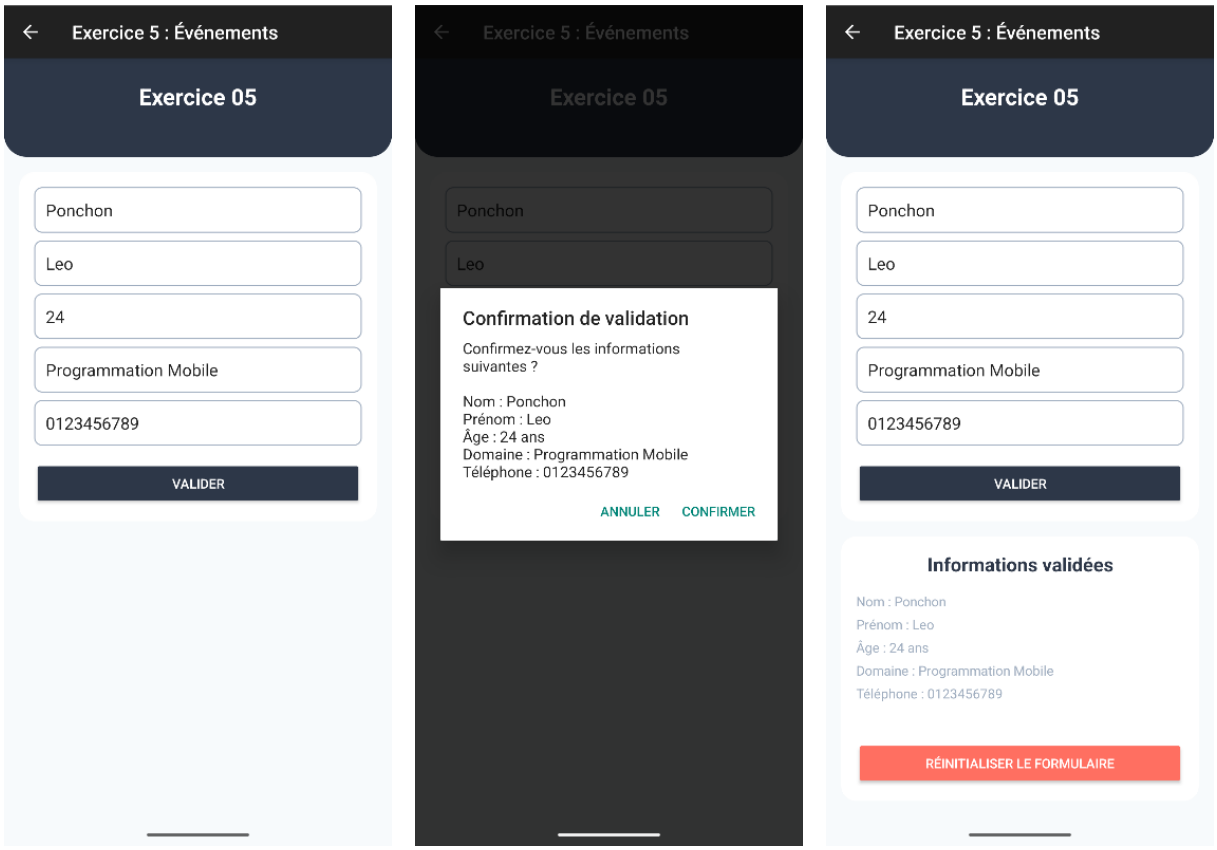

FIGURE 5 – Exercice 5 — Gestion des événements et AlertDialog

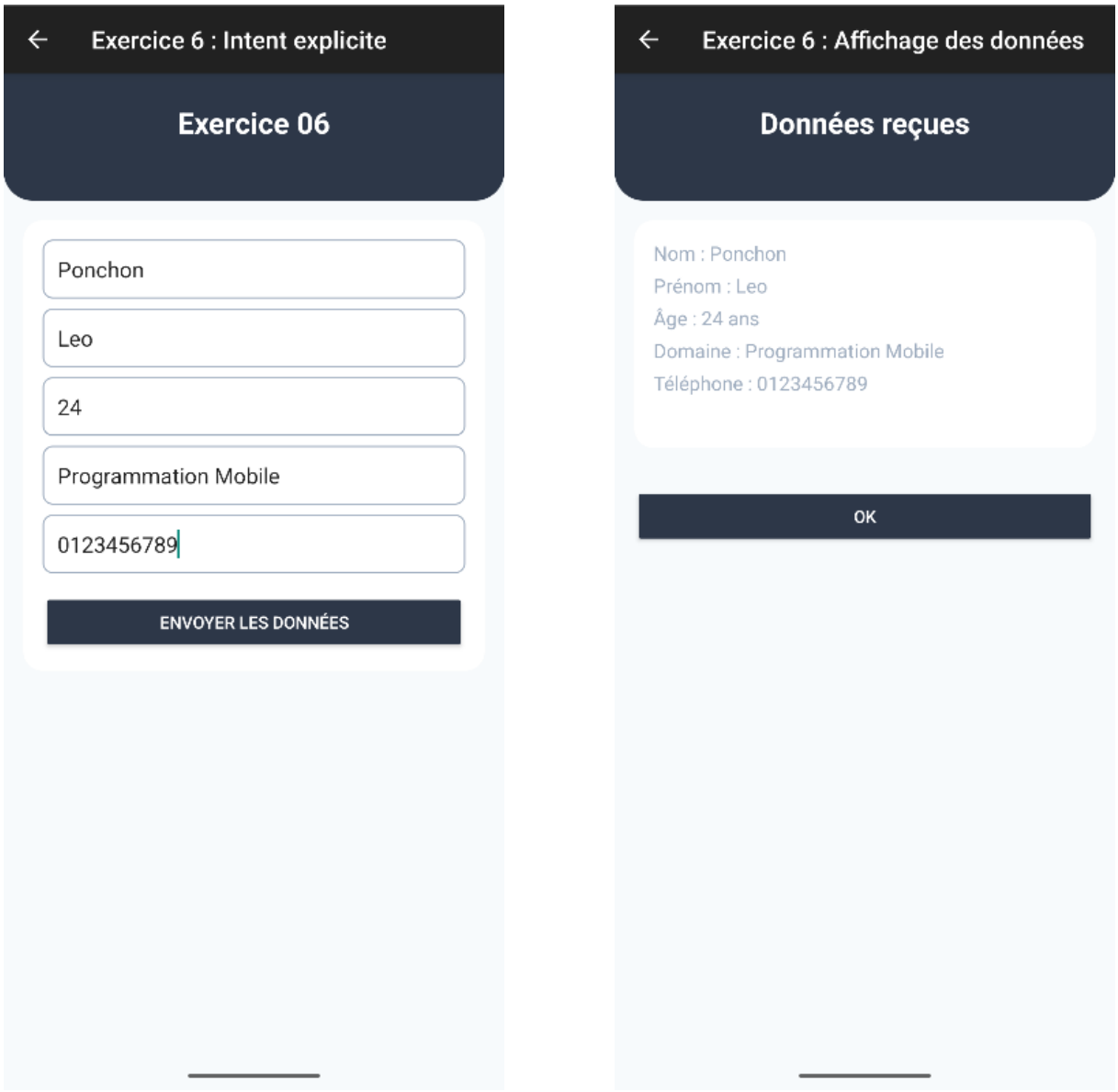

FIGURE 6 – Exercice 6 — Navigation entre activités avec intent explicite

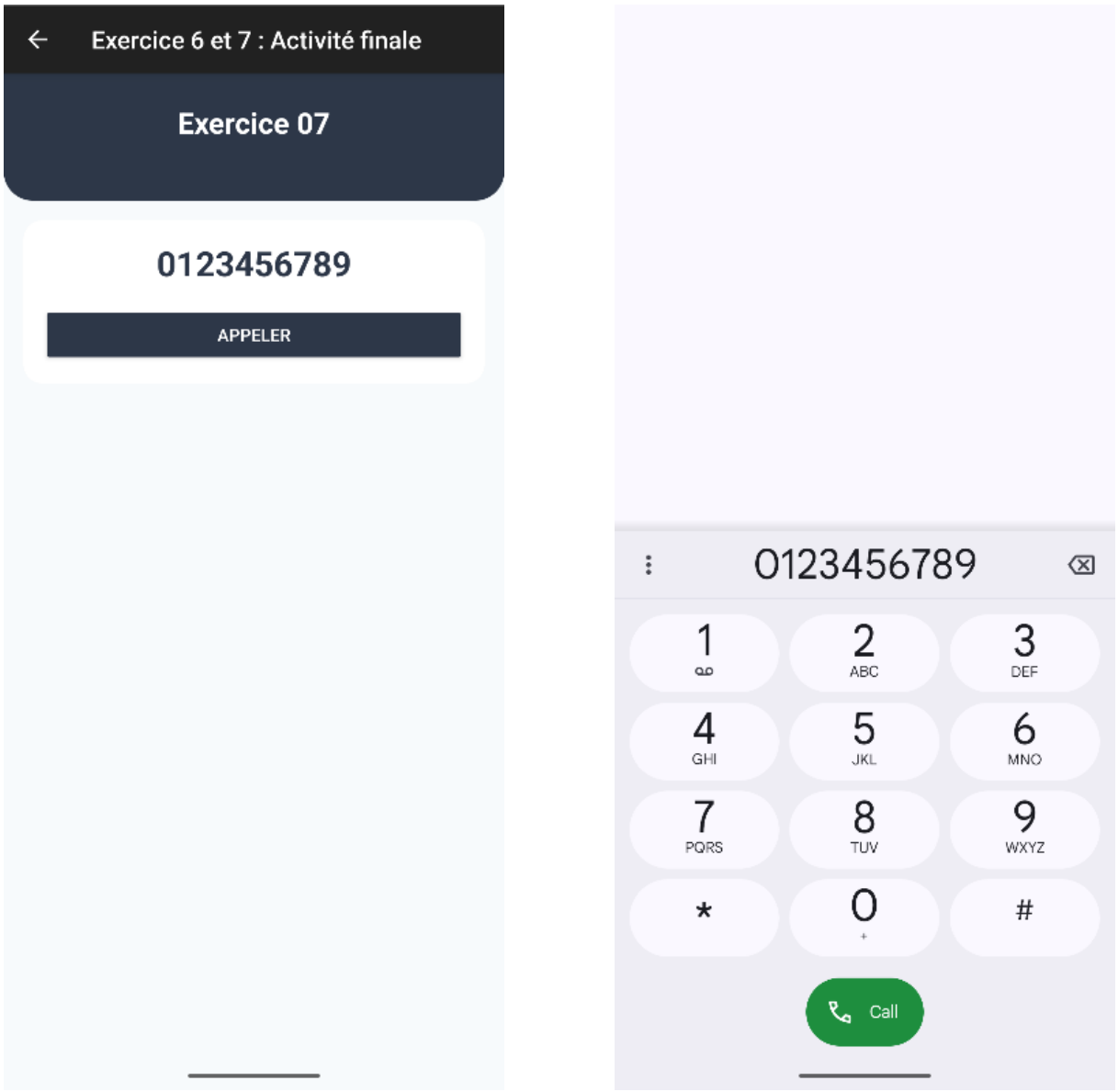

FIGURE 7 – Exercice 7 — Intent implicite pour appel téléphonique

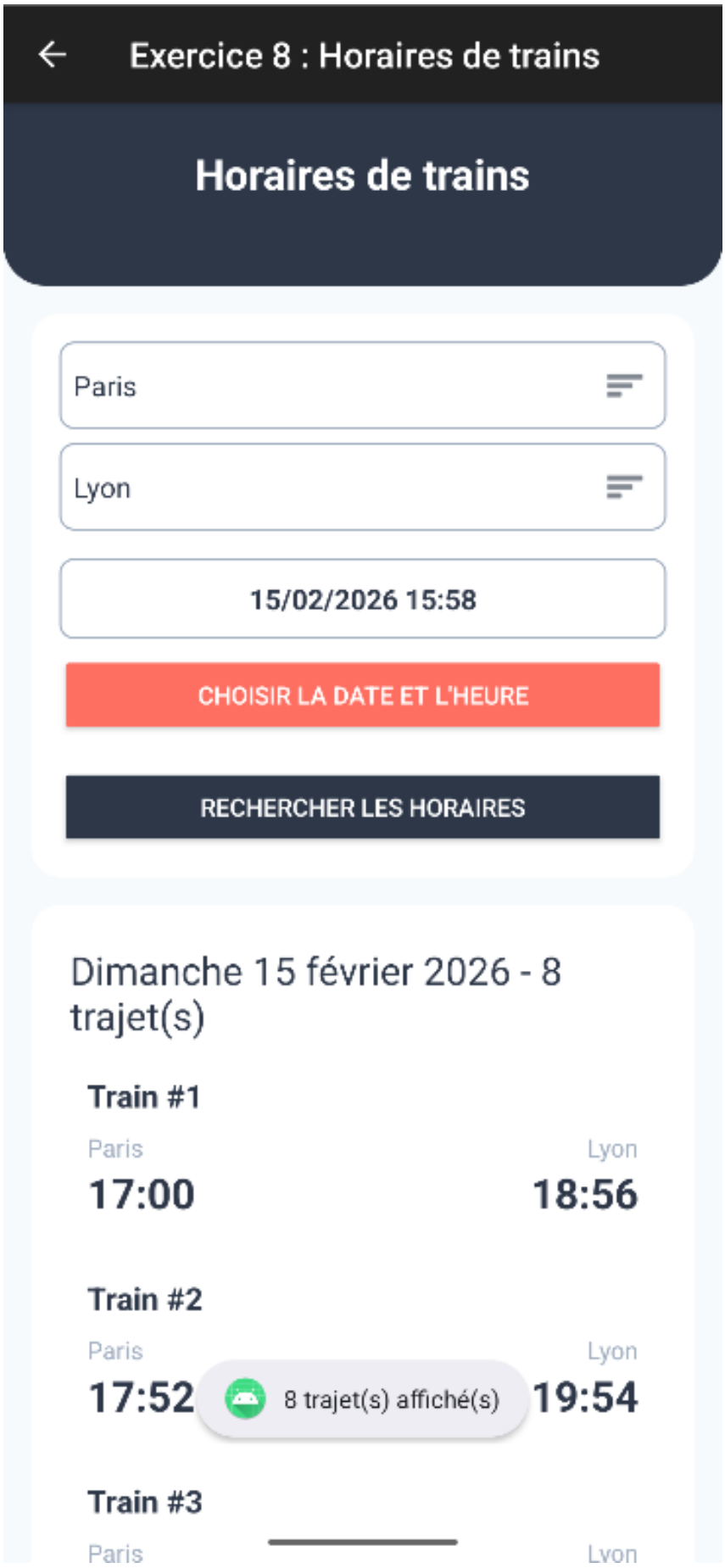

FIGURE 8 – Exercice 8 — Horaires de trains (API SNCF)

FIGURE 9 – Exercice 9 — Agenda avec gestion d'événements