

Rapport TP1

Développement et programmation pour supports mobiles

Bouallouche Yanis | Bengoufa Ryan

Sommaire

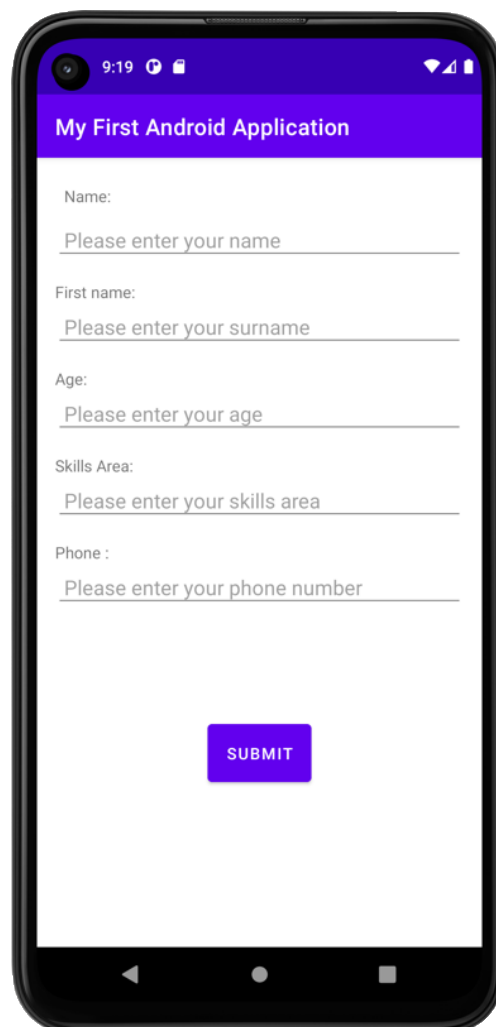
<i>Exercice 3 à 7</i>	<u>3</u>
<i>Exercice 8</i>	<u>6</u>
<i>Exercice 9</i>	<u>9</u>

Exercice 3 à 7

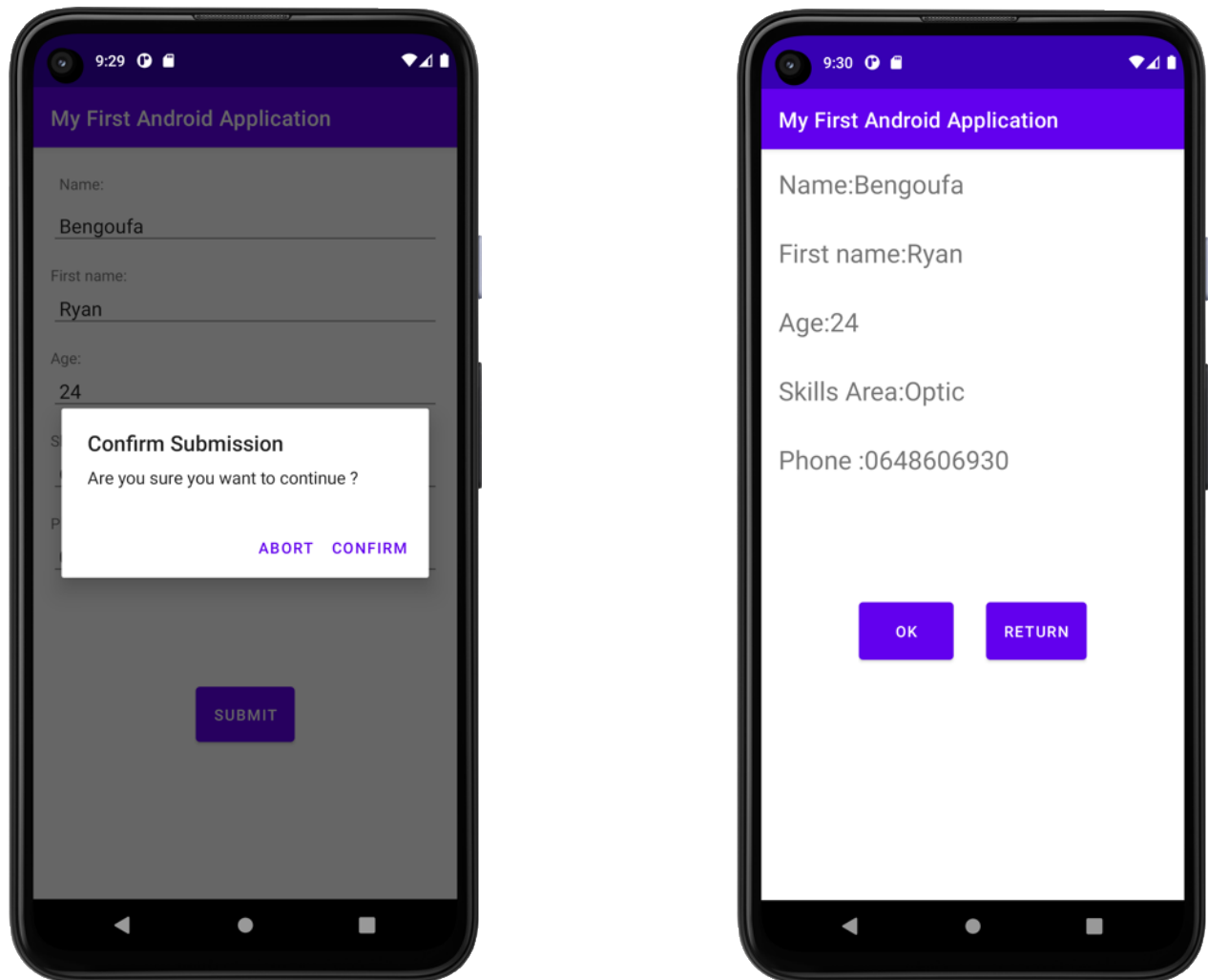
Grâce à ces exercices nous avons pu apprendre les bases du développement d'une application Android, nous avons d'abord mis en place une vue simple qui demande à un utilisateur des informations tels que le nom, prénom, compétences...

Nous l'avons fait en java dans un premier temps puis en XML dans un second temps, nous avons pu remarquer que séparer la logique métier de l'interface graphique est plus agréable notamment si l'application devient plus enrichie. Cela nous permet d'avoir une meilleure organisation.

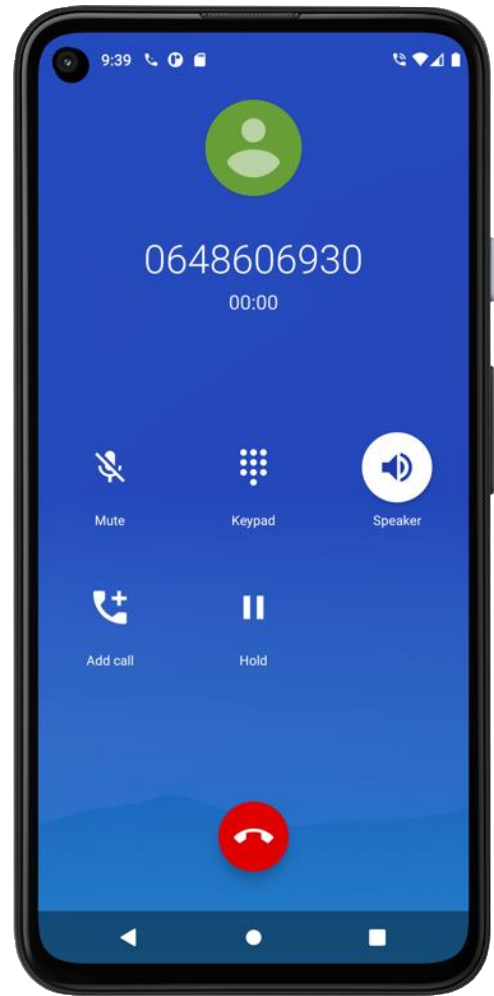
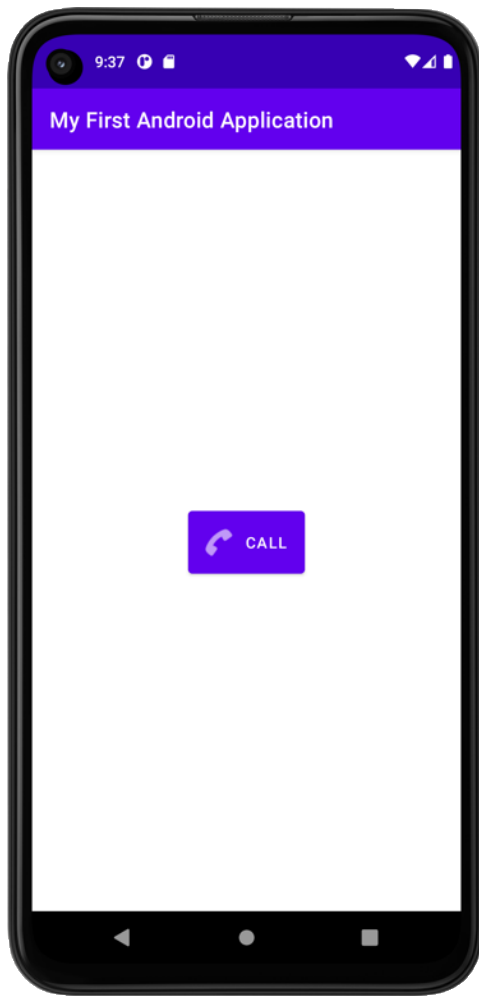
Nous avons ensuite introduit l'internationalisation de l'application grâce aux fichiers **strings.xml** en une version française et une version anglaise, l'application d'adapte automatiquement au langage paramétré dans l'appareil.



L'utilisateur doit saisir tous les champs pour appuyer sur le bouton de validation, une fois les informations saisies, une fenêtre de dialogue s'ouvre pour demander à l'utilisateur s'il souhaite valider ça saisie et continuer.



Une fois que l'utilisateur valide ses informations, grâce à une Intent nous recuperons les données, et lançons une nouvelle activité qui résume les données saisies dans les champs plustôt. Le bouton **RETURN** nous permet de revenir à l'activité précédente, et le bouton **OK** lance une nouvelle activité avec un bouton qui permet CALL qui permet d'appeler le numéro saisi dans la première activité.

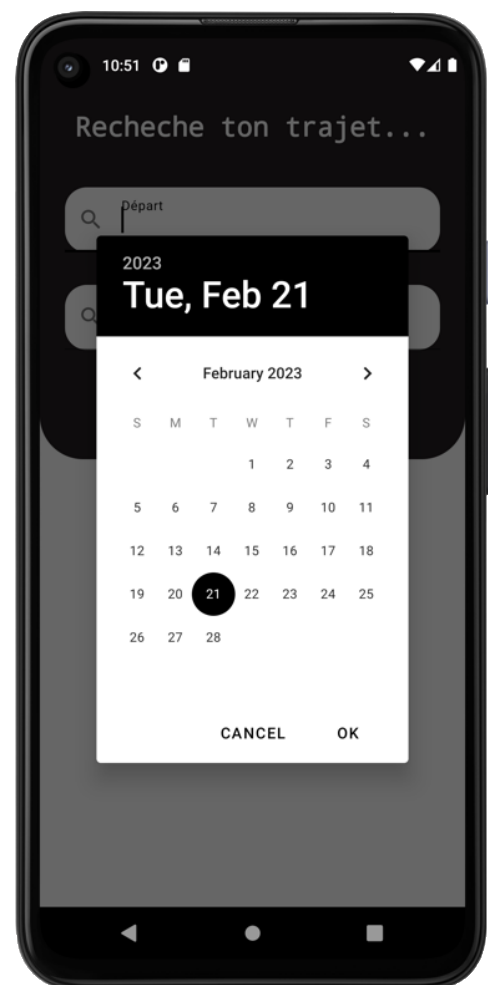


Exercice 8

Dans cet exercice, il nous est demandé de développer une interface graphique permettant à un utilisateur de saisir un itinéraire (villes Départ et Arrivée) et de visualiser tous les horaires de trains pour cet itinéraire sous forme de liste.

Nous avons mis en place une activité qui propose comme demandé à l'utilisateur de saisir les villes pour l'itinéraire, il peut aussi choisir une date celle-ci n'est pas prise en compte lors de la recherche c'était surtout pour un côté esthétique, les champs de saisis sont couplés avec un **autocomplete** pour aider l'utilisateur lors de sa recherche et une croix qui reset le champ.

Si nous souhaitons voir tous les trajets disponibles, nous pouvons le faire en cliquant sur le bouton **Voir Tout**.





Pour afficher les trajets disponibles, nous utilisons une liste dynamique, donc pour mettre cela en place, nous avons créé un fichier xml qui représente notre item individuel (la carte qui affiche le trajet) et nous faisons une injection de données et un affichage grâce au concept de RecyclerView.

Pour améliorer, l'utilisation de l'application, nous avons établie vue défilante.



Exercice 9

Dans cet exercice, il nous est demandé de développer uniquement l'interface graphique d'une application permettant de réaliser un agenda c'est-à-dire d'afficher les dates avec leurs événements associés, ajouter des événements à associer à une date, etc...

Quand nous accédons à l'activité principal, nous avons le calendrier simple, si on appuie sur la date du jour, les données créées en dur dans le code Java apparaissent, donc nous avons aussi utilisé un RecyclerView pour afficher les tâches.



Lorsque l'utilisateur appuie sur le bouton **OK** d'une tâche, alors la tâche est finie et on la retire de la liste.



On peut aussi ajouter des tâches, pour cela il faut sélectionner le jour dans le calendrier et appuyer sur le bouton flottant représenté par un **PLUS**.

Donc si on change de jour par exemple le 22 février, il n'y a pas de tâche prévue pour ce jour.

Après avoir appuyé sur le bouton **PLUS**, une boîte de dialogue s'ouvre et propose à l'utilisateur de saisir un nom, une description et une heure pour créer un événement. Dans l'exemple ci-dessous, nous avons créé deux événements pour le 22 février.

