TP1 - Les Bases d'Android

Pour les exercices d'avant, nous les avons vu en cours avec le professeur, ainsi nous discuterons des exercices suivant.

Exercice 3 : Une première application - Interface simple

Dans les fichiers SaisieActivity.java et activity_saisie.xml.

Figure 1: Activité de saisie de coordonnées

Figure 2: Visuel pour remplissage de l'age

Nous pouvons voir sur la figure 1 que nous avons plusieurs champs de saisie pour remplir les coordonnées. "Prénom", "Nom" et "Domaine de compétence" sont des champs de textes alors que "Age" et "Numéro de téléphone" sont des champs de nombre et de téléphone respectivement. Il y a également un bouton "valider" pour que l'utilisateur puisse valider ses informations.

Ainsi, nous avons fait attention dans notre code à ce que tous les champs soient rempli correctement. C'est-à-dire qu'ils soient tous remplis d'au moins un caractère, et que le numéro de téléphone est au moins 10 caractères (en supposant que ce soit un numéro français).

Par ailleurs, nous avons également ajouté un bouton "Accueil" pour revenir au menu qui est un affichage des 3 onglets de l'application (saisie, train et agenda).

Nous pouvons donc voir sur la figure 15 que le clavier est adapté au champ de saisie que l'on remplis. Ainsi, pour l'âge ce sont des chiffres qu'il faut remplir.

Exercice 4 : Internationalisation des interfaces

Dans les fichiers MainActivity.java et activity_main.xml.

Grace au bouton en haut à droite, il est possible de changer directement la langue d'affichage sans passer par les paramètres du téléphone. Ici, comme nous avons deux langues, il suffit d'intervertir les langues ; si la page est en anglais (figure 4 de la page 2), l'application passe en français et inversement (figure 3 de la page 2). Pour ce faire, nous avons le fichier string.xml en français, et un autre en anglais dans les dossiers /res.

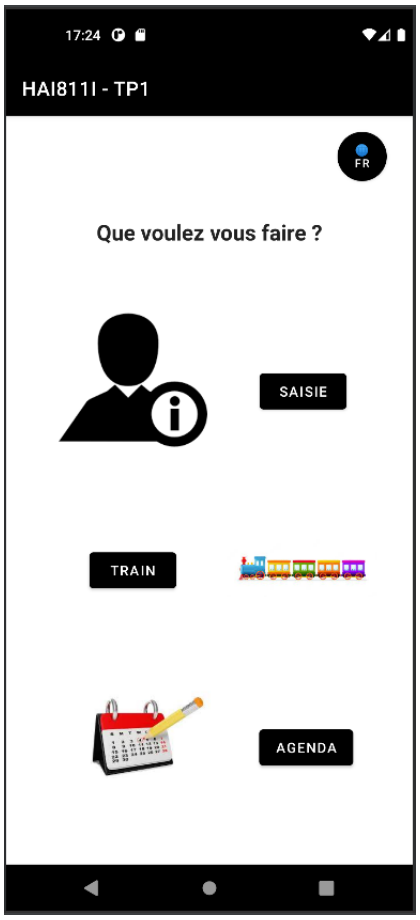

Figure 3: Activité d'accueil en français

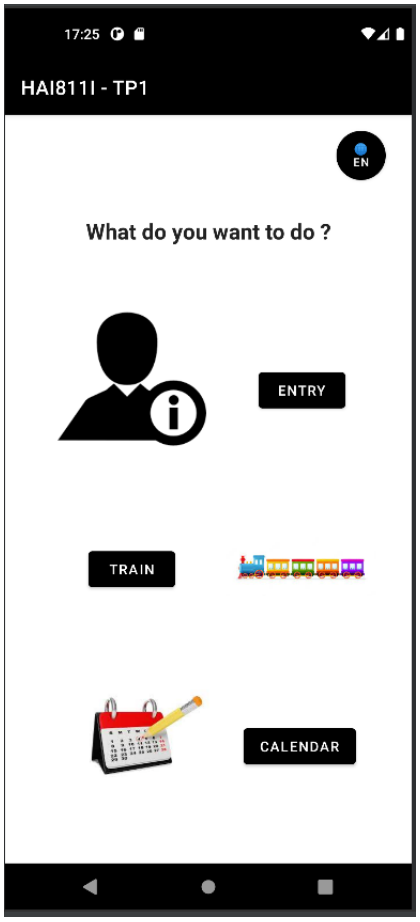

Figure 4: Activité d'accueil en anglais

Exercice 5 : Événements associés aux objets graphiques d'une vue
Exercice 6 : Intent explicite
Exercice 7 : Intent implicite

Dans les fichiers AffichageActivity.java, AppelActivity.java et activity_saisie.xml, activity_appel.xml.

Nous voyons sur la figure 5 à la page 3, que lorsqu'on appuie sur le bouton valider, on ouvre une boite de dialogue qui nous demande si on veut valider les informations remplies ou non.

Soit l'utilisateur dit "non", et il ré-accède au formulaire de saisie.

Soit il dit "oui", et il accède à l'affichage de nos données sur une nouvelle activity montré sur la figure 6 à la page 3.

On a également sur chaque activity permis de revenir en arrière avec un bouton retour cette fois-ci pour retourner seulement à la saisie des données.

Lorsque l'utilisateur appuie sur le bouton "ok", ça l'emmène à une autre activity, celle sur la figure 7 à la page 4, où on voit le numéro de téléphone saisi et la possibilité d'appeler ce numéro.

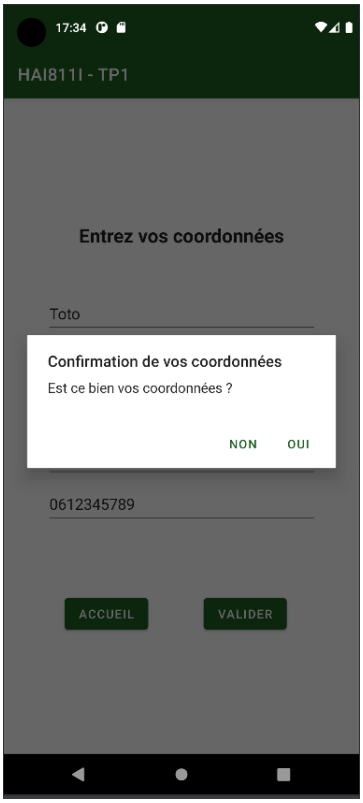

Figure 5: Boite de dialogue après validation

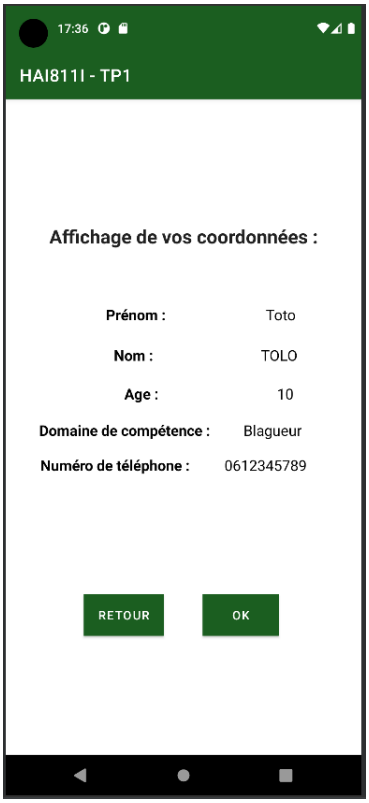

Figure 6: Affichage des données

Et pour finir lorsque l'utilisateur clique sur le bouton "appeler", cela appel le numéro à l'aide du compositeur de téléphone Android. C'est ce qu'on peut voir sur la figure 8 à la page 4.

Exercice 8 : Application simple pour consulter les horaires de trains

Pour le train, nous avons affiché deux editText pour remplir la ville de départ et la ville d'arrivé avec un bouton "rechercher" (et un bouton accueil), ainsi l'interface graphique est montré à la figure 9 à la page 5. Ce qui affiche comme on peut le voir sur la figure 10 à la page 5, les différents trajets possibles entre les deux villes remplis.

Les différents trajets sont calculées aléatoirement parmi une liste prédéfinie de ville existante, car nous avons fait en sorte que les trajets soient générés en se limitant aux villes suivantes : Montpellier, Béziers, Toulouse, Paris, Lyon, Grenoble, Valence et Versaille. Un utilisateur qui chercherait un trajet aberrant comme "rfbfisg" vers "ndrgiwg" n'obtiendrais donc aucun résultat. On peut voir un exemple sur la figure 11 à la page 5 lorsqu'on a volontairement fait une faute de frappe à Grenoble.

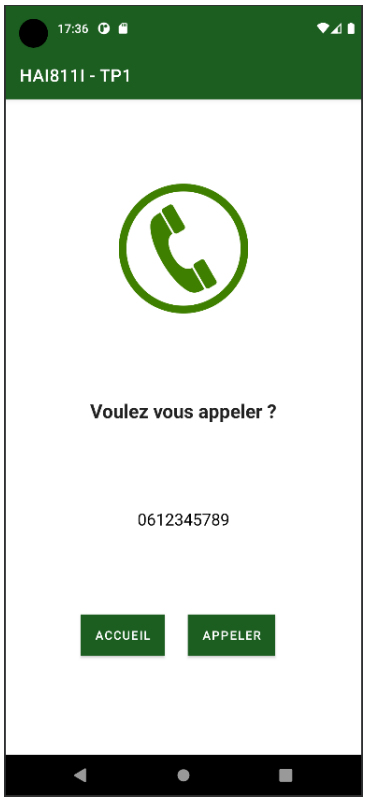

Figure 7: Affichage intermédiaire avant l'appel

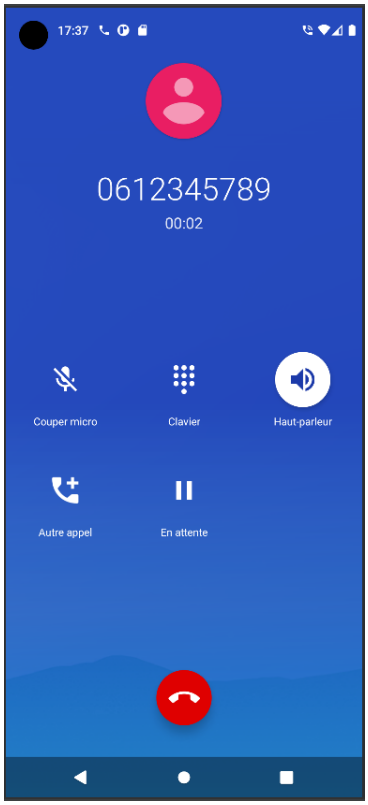

Figure 8: Appel du numéro

Exercice 9 : Application simple d'agenda

Pour cet agenda, nous avons utilisé un CalendarView qui permet d'afficher le calendrier en entier en choisissant le premier jour, l'intervalle de temps proposé (avec des bornes). Ainsi, a chaque sélection d'une date on affiche grâce à une listView les événements de ce jour, c'est donc ce qu'on peut voir sur la figure 12 à la page 6. Nous avons ajouté "Bonjour ! Nous sommes le 20/01/2023 sur la date du jour que nous pouvons voir sur la figure 13 à la page 6.

Pour ajouter un événement au calendrier, il suffit de rentrer le nom de l'événement dans le champ "Nouvel événement", puis de cliquer sur le bouton "ajouter" ; l'événement sera alors ajouté pour la date du jour (figure 15 et figure 16 à la page 5).

De plus, nous avons également ajouté la possibilité de supprimer un événement du calendrier. Pour ce faire, il suffit de cliquer sur ce dernier, et une boite de dialogique apparaît pour laisser le choix à l'utilisateur de supprimer ou non l'événement on peut le voir sur la figure 14 à la page 6.

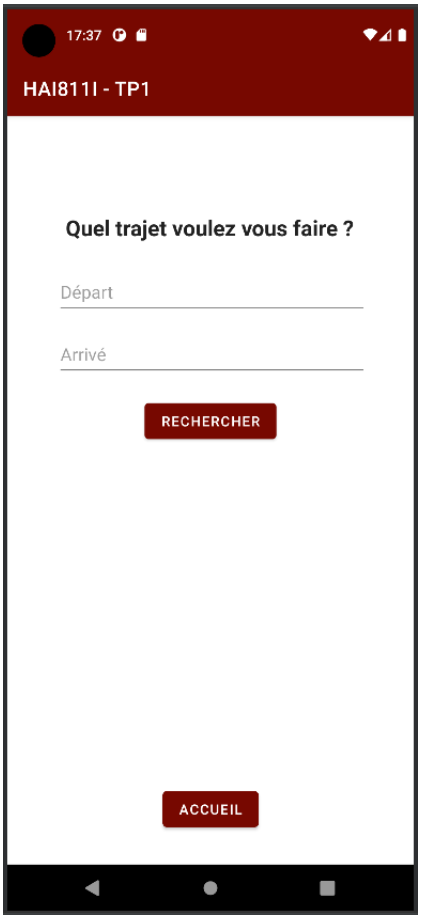

Figure 9: Affichage de l'accueil

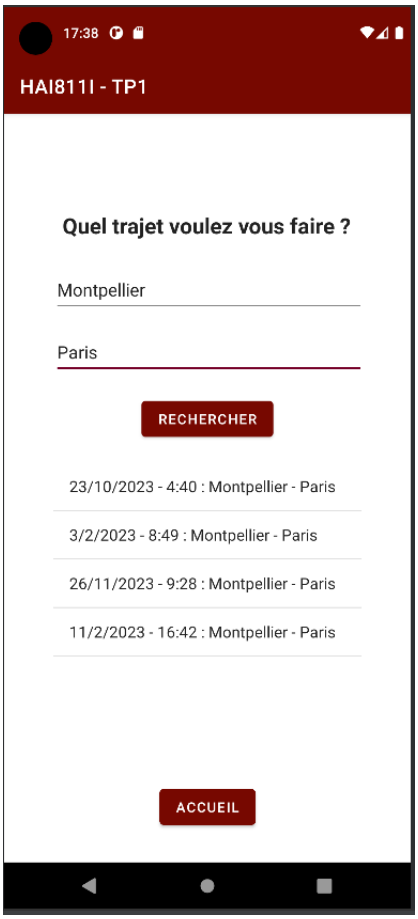

Figure 10: Affichage de la liste des trains

Figure 11: Affichage lorsque pas de train

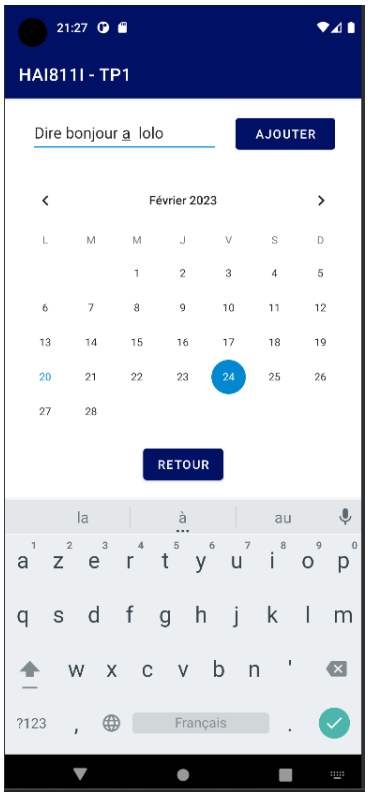

Figure 15: Ajout d'un évènement

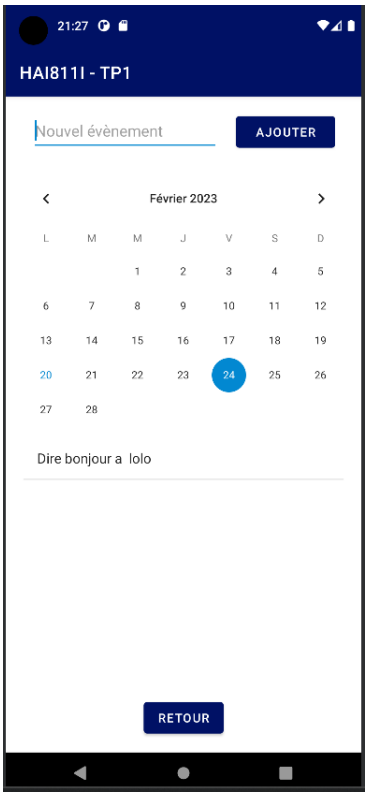

Figure 16: Evenement ajouté

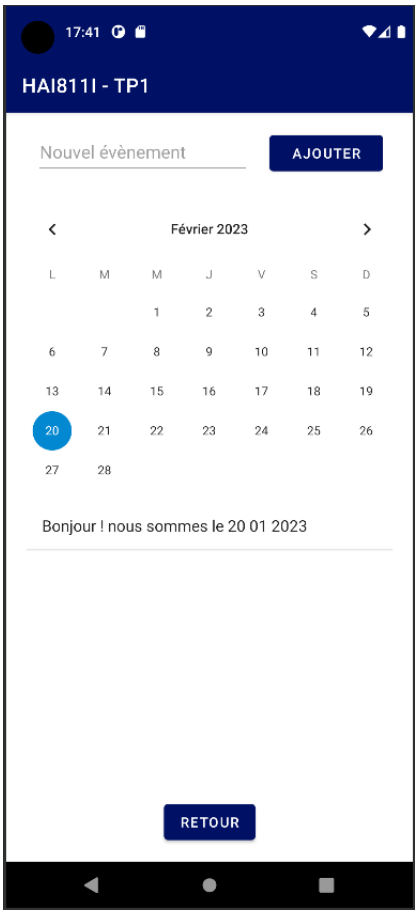

Figure 12: Affichage du calendrier sur la date d'aujourd'hui

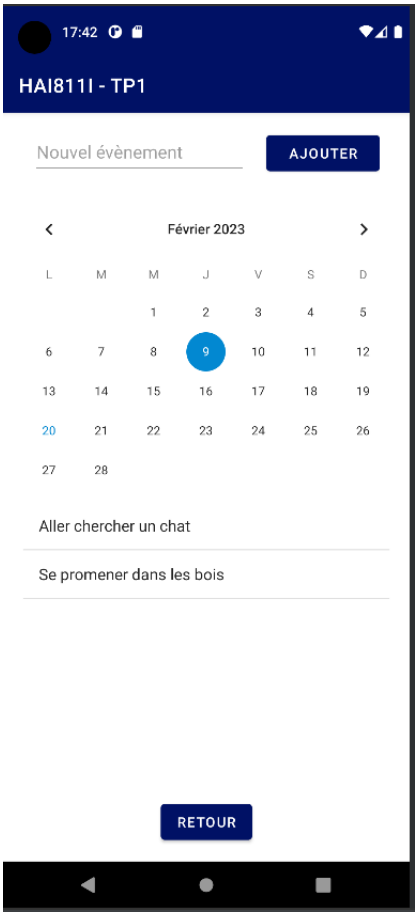

Figure 13: Affichage après avoir ajouté un élément sur une autre date

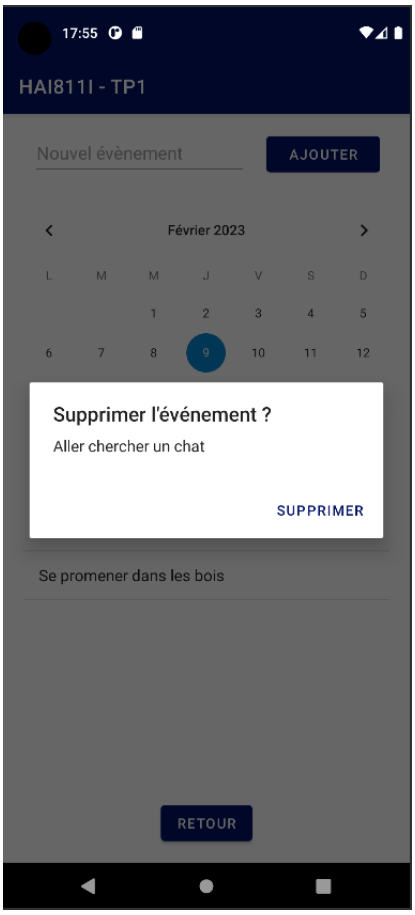

Figure 14: Boite de dialogue pour la suppression de l'événement

Fonctionnalités supplémentaires

Nous avons ajouté une page d'accueil que nous avons vu plus haut, qui affiche le bouton de changement de langue en haut à droite comme nous l'avons vu plus haut. Une image et un bouton pour chaque application du TP, à savoir, une saisie de données, un affichage de train et pour finir un agenda où on peut ajouter et supprimer des événements.

Une des améliorations possibles serait d'améliorer les interfaces graphiques des listView qui sont actuellement assez simple.