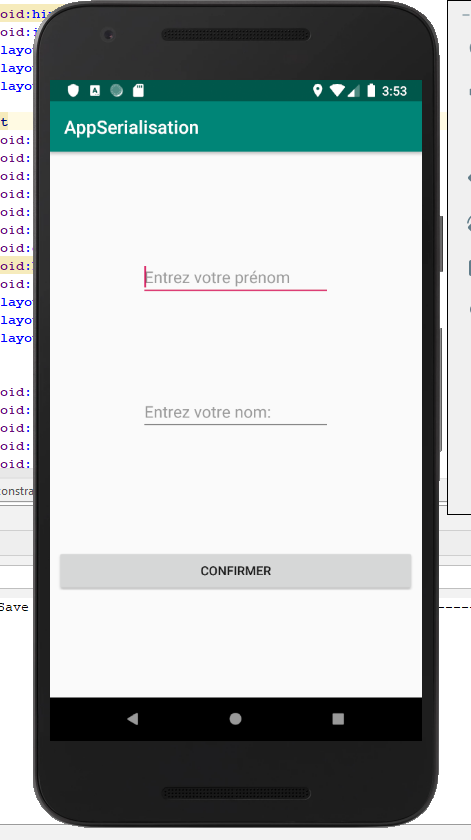
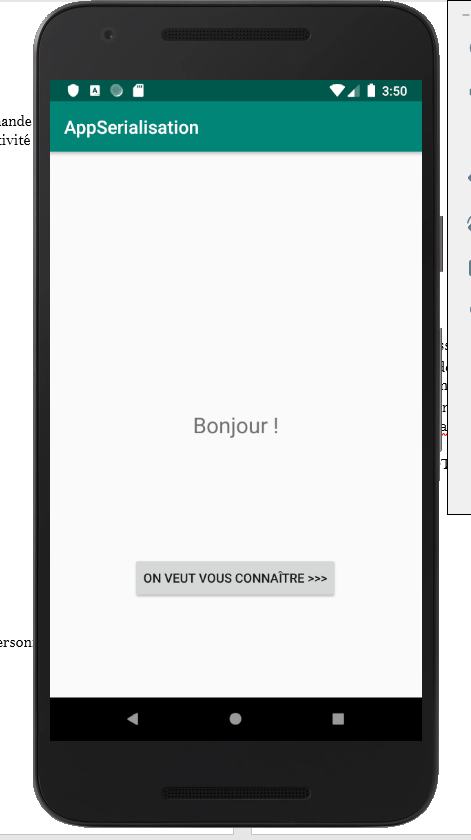
# Annexe 4 – startActivityForResult, saveInstanceState

Nouveaux éléments :

* startActivityForResult ( regarder projet ImportanceBufferMP3 Annexe 2 pour vous donner une idée )
* saveInstanceState

# A)Exercice :

Une application simple où on salue l’usager dans une Activity; on lui demande d’entrer des informations sur lui dans une seconde activité afin de revenir sur l’activité initiale avec une salutation personnalisée :



Quand on clique sur confirmer, on revient sur la première activité en personnalisant le message d’accueil avec les entrées texte:



* Utiliser le projet où j’ai déjà dessiné les interfaces; codez les gestions d’événements clic
* Coder une petite classe Utilisateur dans son propre fichier .java contenant comme attributs un prénom et un nom ; il s’agira des informations qu’on demandera à l’usager par le « boomerang ».
* Étant donné qu’on est en présence d’un « boomerang », on va reprendre la technique « startActivityForResult » qu’on a utilisée à l’annexe 2. Suivez les indications suivantes ( en vous inspirant du travail de l’annexe 2 pour la 1ère partie du moins ) :

## Nouvelle technique ( startActivityForResult )

**Activité de départ ( AccueilActivity ) :**

* + **Déclarer un objet ActivityResultLauncher<Intent>**
  + **Le créer ( dans le onCreate ) à l'aide de la méthode registerForActivityResult contenant 2 paramètres :**
    - **Un objet de type ActivityResultContracts.StartActivityForResult()**
    - **Un objet d'une classe** **mettant en œuvre l'interface ActivityResultCallback<ActivityResult>**
  + **Lancer un intent avec la méthode launch sur notre objet ActivityResultLauncher<Intent> lorsqu’on clique sur le bouton**

**Activité de destination ( IdentificationActivity )**

* **créer un Intent de retour sans destination ( new Intent() )**
* **gérer le clic du bouton qui permettra de transmettre l’objet Utilisateur à l’activité de départ ( retour du boomerang )**
* **putExtra( « identifiant », objet Utilisateur ); moyen pour transmettre des infos d’une activité à l’autre par un Intent**
  + **comment passer un objet Utilisateur comme extra ??**
  + **Comment rendre une classe / objet sérialisable ?? en implémentant l'interface Serializable ( marker interface )**
* **setResult(code de retour , nom de l’intent de retour )**
* **finish()**

**Activité de départ ( AccueilActivity )**

* **notre intent de retour ( boomerang ) revient dans la méthode onActivityResult de la classe interne mettant en œuvre l'interface ActivityResultCallback<ActivityResult>**
* **vérifier si le code de retour est le bon**
* **récupérer l'intent puis l'extra à partir du paramètre de type ActivityResult**

Ca fonctionne ?

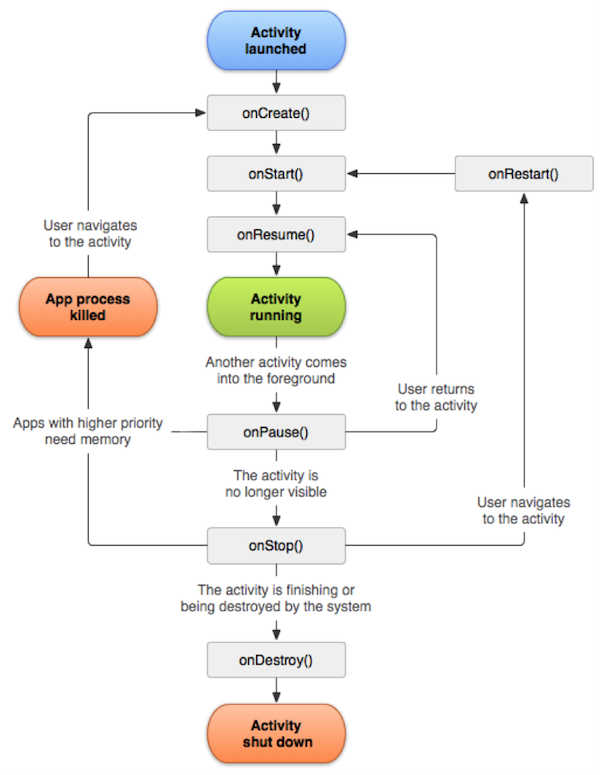
Dans les paramètres du téléphone, changer dans Settings 🡪 Display 🡪 Auto-rotate screen à true

Tournez l’émulateur que se passe-t-il avec le nom de l’usager ? Cliquez ou appuyez ici pour entrer du texte.

Pourquoi ? Cliquez ou appuyez ici pour entrer du texte.

Cliquez ou appuyez ici pour entrer du texte.

# B)cycle de vie



onRestoreInstanceState est appelé

Tourner le téléphone

retour d'avoir changé la langue du téléphone, …

Retour du bouton Recents, du bouton Home, "retour du boomerang"

onSaveInstanceState est appelé

Recoit un appel

Ferme L’app

Attention ! onDestroy n’est pas tjrs appelé selon la doc d’Android

<https://developer.android.com/guide/components/activities/activity-lifecycle>

<https://medium.com/@theabhishekavi/android-activity-lifecycle-9bc7de812dff>

**A-Solution : Si on veut garder les infos durant la session ou le processus courant : savedInstanceState**

* La méthode **protected void** onSaveInstanceState(Bundle infos) est appelée juste avant le onStop dans le cycle de vie d’une Activité; de cette façon, on peut sauvegarder des infos importantes lorsque le cycle de vie est refait au complet pcq on a tourné le téléphone, entre autres. Cependant, lorsque le processus de l'activité est complètement terminée, on perd les infos présentes dans le Bundle
* Permet donc de conserver les infos quand on demeure dans la même session / le même processus
* Utilisé pour conserver un peu d'informations
* On va ajouter au Bundle en paramètre notre Utilisateur en extra. Un Bundle est un ensemble d'informations "extras"

Méthodes : putSerializable :

getSerializable :

* Vu que la rotation du téléphone entraîne qu’on repasse dans le onCreate, on va récupérer notre extra dans le paramètre savedInstanceState

Ca fonctionne ?

Finalement, est-ce qu’on doit passer à travers toutes ces étapes pour chaque app où l’usager tournera le téléphone ou changera la langue de son téléphone

Bien sûr que non ! : Truc : Modifier le fichier manifest pour dire que l’activité ne se redémarre pas si on tourne le téléphone ( orientation, autres situations possibles .. )

**android:configChanges="orientation|screenSize|keyboard|keyboardHidden"**

B- **Solution : Si on veut garder les infos au-delà de la fermeture de la session ou du processus courant : sérialisation de l’objet, utilisation de fichiers texte ou d’un table d’une BD.**

Qu’arrive-t-il si on ferme l’application sur le téléphone ?? On est plus ds la même instance donc on perd le prénom/ nom pour de bon

**Façons de réellement terminer une Activité / une instance de cette activité ( Android 12 ou plus ) ?**

Cliquez ou appuyez ici pour entrer du texte.

Pratiquer le processus de sérialisation pour conserver le nom de l’utilisateur même s’il ferme l’application

Où sérialiser ? Cliquez ou appuyez ici pour entrer du texte.

Où désérialiser ? Cliquez ou appuyez ici pour entrer du texte.