



SMB116 : Les Ressources

TP4 : Adaptation d'une application en fonction de la locale

Partie 1 :

1.1) Créer une activité permettant d'envoyer des SMS et des Emails. Ces fonctionnalités seront implémentées en utilisant les composants préexistants sur le dispositif Android.

L'application comportera une seule activité :



Le bouton « Enveloppe » permettra d'ouvrir la liste des contacts pour sélectionner le destinataire puis de composer un mail pour ce destinataire.

Le bouton « Clavier » permettra d'ouvrir la liste des contacts pour sélectionner le destinataire puis de composer un SMS pour ce destinataire.

Le bouton quitter permet de quitter l'activité et de fermer l'application

La manière d'utiliser les applications de Mail et SMS de l'équipement Android est fournie par cette documentation :

<https://developer.android.com/guide/components/intents-common#java>

Layout :

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TableLayout
        android:layout_width="368dp"
        android:layout_height="231dp"
        tools:layout_editor_absoluteX="8dp"
        tools:layout_editor_absoluteY="8dp">

        <TableRow
            android:layout_width="match_parent"
            android:layout_height="match_parent" >

            <ImageButton
                android:id="@+id/btnMail"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                app:srcCompat="@android:drawable/ic_dialog_email"
                android:layout_weight="1"/>

            <TextView
                android:id="@+id/lblAdresse"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="TXT_ADRESSE_MAIL"
                android:layout_weight="5"/>
        </TableRow>

        <TableRow
            android:layout_width="match_parent"
            android:layout_height="match_parent" >

            <ImageButton
                android:id="@+id/btnSms"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                app:srcCompat="@android:drawable/ic_dialog_dialer"
                android:layout_weight="1"/>

            <TextView
                android:id="@+id/lblNumeroTelephone"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="TXT_NUMERO_TELEPHONE"
                android:layout_weight="5"/>
        </TableRow>

        <TableRow
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            >

            <Button
                android:id="@+id/btnQuitter"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="Quitter"
                android:layout_span="2"
                android:layout_weight="1"
                android:drawableLeft="@android:drawable/ic_lock_power_off" />

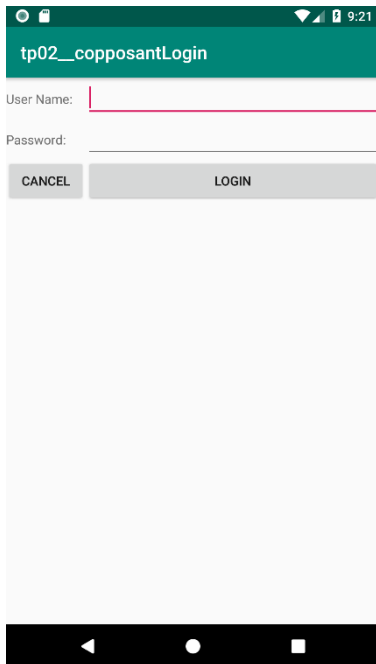
        </TableRow>

    </TableLayout>

</android.support.constraint.ConstraintLayout>
```

Partie 2 :

2.1) Créer un nouveau module (ou application) de « Login ». Il s'agira d'un composant pouvant être utilisé par d'autres applications pour demander à leur utilisateur de s'authentifier.



Ce composant (son activité) devra pouvoir être utilisé par les autres applications de deux manières :

- **Par l'utilisation d'un Intent Explicite**
- **Par l'utilisation d'un Intent implicite correspondant à l'action « LOGIN_ACTION »**

Dans les 2 cas, il devra être possible à l'application utilisatrice de fournir une valeur par défaut de « UserName » et de « Password ». Ces valeurs seront affectées aux EditText avant l'affichage de l'activité.

Ce composant ne pouvant pas être utilisé de manière autonome, vous supprimerez la possibilité de la lancer comme une application standard à partir du bureau.

Layout :

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".LoginActivity">

    <TableLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:stretchColumns="1">

        <TableRow>
            <TextView
                android:text="User Name: "
```

```

        android:id="@+id/lblUserName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </TextView>

    <EditText
        android:text=""
        android:id="@+id/txtUsername"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">
    </EditText>
</TableRow>

<TableRow>
    <TextView
        android:text="Password: "
        android:id="@+id/lblPassword"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </TextView>

    <EditText
        android:text=""
        android:id="@+id/txtPassword"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword">
    </EditText>
</TableRow>

<TableRow>
    <Button
        android:text="Cancel"
        android:id="@+id/btnCancel"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">
    </Button>

    <Button
        android:text="Login"
        android:id="@+id/btnLogin"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">
    </Button>

</TableRow>
</TableLayout>

</android.support.constraint.ConstraintLayout>

```

2.2) Utiliser ce composant pour s'authentifier au lancement de l'application de communication de la partie 1.

Partie 3 Utilisation des BroadcastReceiver :

3.1) On va ajouter à l'application de communication une fonction qui lorsque la batterie est considérée comme faible prévient avec une boîte de message avec un simple bouton OK qui ferme l'application (voir `AlertDialog.Builder` avec le bouton OK).

3.2) Ajouter une activité accessible à partir d'un bouton « Configuration » qui permet de paramétrer en % le niveau de batterie à partir duquel déclencher l'alerte. Cette valeur devra être mémorisée pour être réutilisée à chaque lancement de l'application.

<https://developer.android.com/training/monitoring-device-state/battery-monitoring>

Partie 4 Utilisation des Permissions :

Pour pouvoir utiliser le composant de login de la « Partie 2 » l'application de communication doit obtenir la permission : « `comlogin.permission.LOGIN` ».

**Si l'application n'a pas cette permission elle devra la demander à l'utilisateur en interactif.
Si l'utilisateur la refuse, l'application devra s'arrêter après un message d'avertissement.**

Partie 5 Utilisation des Ressources :

5.1) Votre super composant de Login va être vendu dans des pays anglophones et au Japon. Il est donc indispensable d'internationaliser les labels. Les traductions suivantes vous ont été fournies :

Français	Anglais	Japonais
Code Utilisateur	User Name	ユーザー名
Mot de passe	Password	パスワード
Annuler	Cancel	キャンセル
Connexion	Login	ログイン

Il vous est demandé d'internationaliser votre composant pour permettre son utilisation en langue Française, Anglaise et Japonaise.

5.2) Votre super composant de Login doit s'adapter à la culture du pays. D'après ce que vous a dit le directeur commercial pour le Japon : « Les Japonais ont l'habitude d'avoir le mot de passe affiché en premier et le code utilisateur en second lorsque l'écran est en orientation paysage, en orientation portrait c'est l'inverse ».

Il vous est demandé d'adapter l'organisation de l'écran aux habitudes des Japonais.