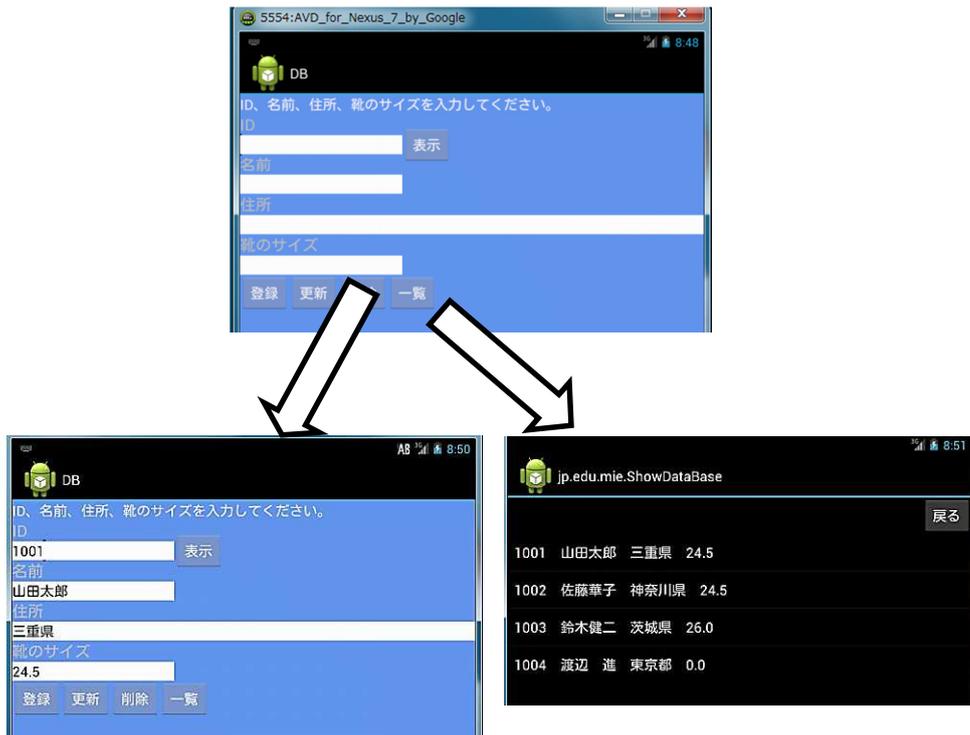


2 靴屋さんの顧客管理データベースの概要

靴屋さんの顧客管理を行うためのプログラムをリレーショナルデータベースを使って作ります。

- ・ ID 番号（顧客管理用の番号）
- ・ 名前
- ・ 住所
- ・ 靴のサイズ（cm）

「登録」「更新」「削除」「一覧」の各機能を備えています。



3 文字列定数を定義する。

ファイル名： res/Value/strings.xml (新規作成)

```
<resources>
  <string name="app_name">DBCcustomerCard</string>
  <string name="label_title">ID、名前、住所、靴のサイズを入力してください。</string>
  <string name="label_id">ID</string>
  <string name="label_name">名前</string>
  <string name="label_address">住所</string>
  <string name="label_size">靴のサイズ</string>
  <string name="button_insert">登録</string>
  <string name="button_update">更新</string>
  <string name="button_show">表示</string>
  <string name="button_delete">削除</string>
  <string name="button_list">一覧</string>
  <string name="button_return">戻る</string>
</resources>
```

4 ウィジェット(widget)を定義する

4.1 レイアウトの作成

ファイル名: `res/layout/activity_main.xml` (新規作成)

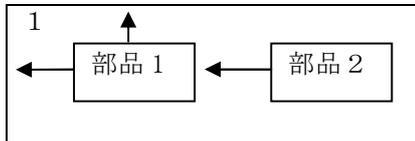
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android" //①
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#6495ED"
    tools:context=".MainActivity" >
    <Button
        android:id="@+id/button_list"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_list" />
</RelativeLayout>
```



①relative layout

relativeLayout は、名前のおり相対的に Widget の配置を行うレイアウトです。自身もしくは、その中に配置された部品を基点に次の部品を配置していきます。その構造は、次のとおりです。

```
<RelativeLayout>
    <GUI 部品名 属性 />
</RelativeLayout>
```



- (1) 部品 1 の配置位置を、RelativeLayout を基点に決定
- (2) 部品 2 の配置位置を、部品 1 を基点に決定

ファイル名: `res/layout/show_database.xml` (新規作成)



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".ShowDataBase" >
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```

        android:layout_gravity="right" //①ボタンの右寄せ
        android:text="@string/button_return" />
    <ListView //②
        android:id="@+id/listView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>

```

①android:layout_gravity="right" ※gravity:重力

layout_gravity を使って、デバイスの画面サイズがわからなくても、部品を上下左右の任意の辺に沿って配置することができます。

LinearLayout では、なにも指定しない場合、垂直方向のレイアウトでは、中の部品は左寄せで、水平方向のレイアウトでは上寄せで配置されます。layout_gravity を指定することで、このデフォルトの寄せの方向を調整することができます。

水平のレイアウトでは、layout_gravity に対し、左 (left)、中央 (center_horizontal)、右 (right) の設定ができます。

垂直のレイアウトでは、layout_gravity に対し、上 (top)、中央 (center_vertical)、下 (bottom) の設定ができます。

②<ListView>

一般的なデータリストを表示するために、ListView という widget が用意されています。ListView はデータを縦方向に並べて表示して、必要に応じてフリックでスクロールします。