Android development course – Exercise 2

Exercise goals:

Practice creation of basic android application that use these basic capabilities:

- Activities
- Menus
- Intents

Instructions:

- Each question descries an Android app to create
- Each question descries the required app behavior
- Each question contains input and required output examples
- Solution examples will be provided next week
- As the questions don't depend on each other, if you encounter any problems solving a question it is recommended to move on to the next one.

Tip: Most of the tools and techniques you need in order to solve these questions we've covered in the class session. Use the code examples seen and reuse and alter them as needed. And as always – a quick Google search might help resolve any issue!

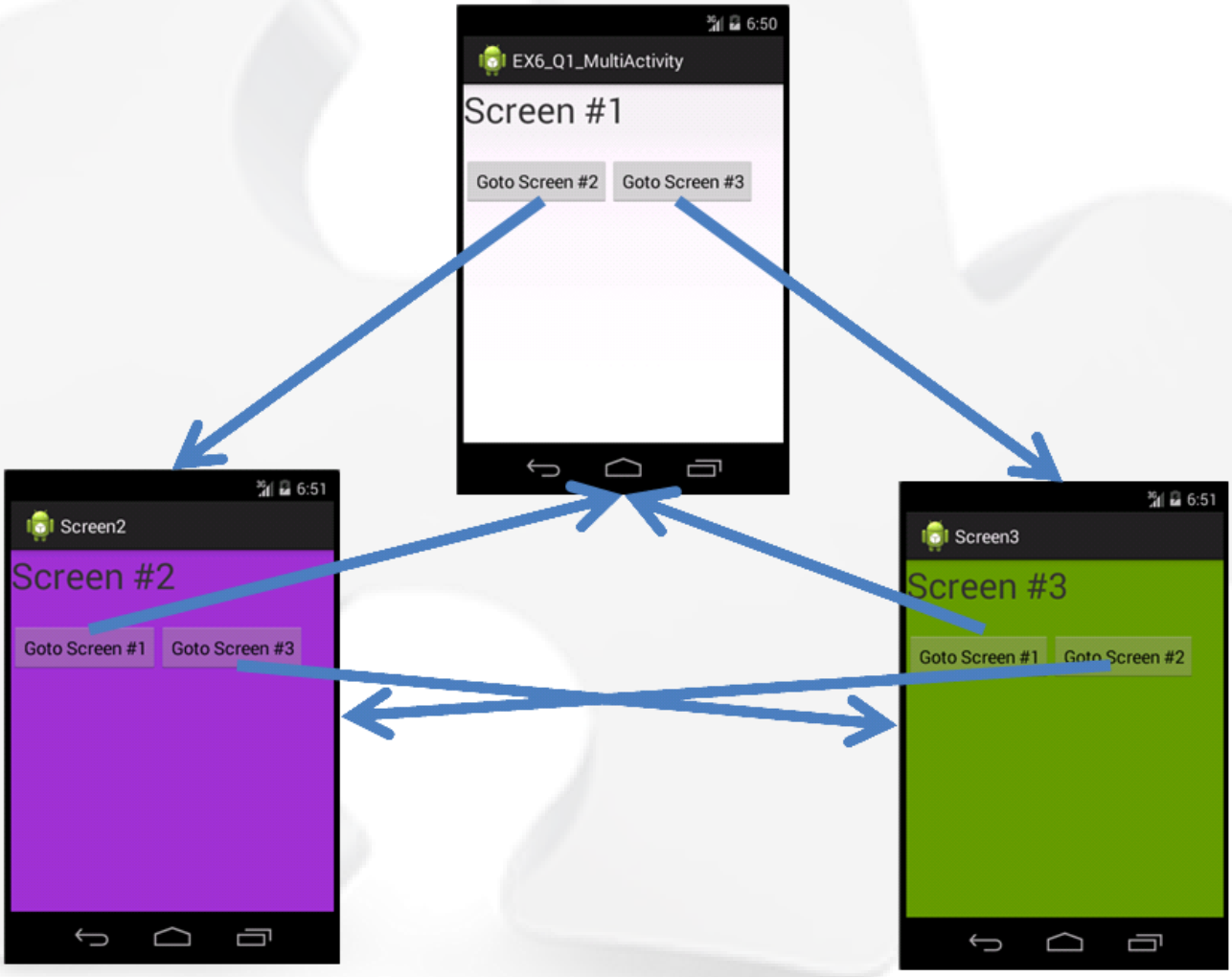
Activities

Question 1 –

Write an app that contains three different screens (create three Activities).

Each screen will have two buttons that navigate to the other screens.

Example:



Question 2 –

Write an app that has several fields:

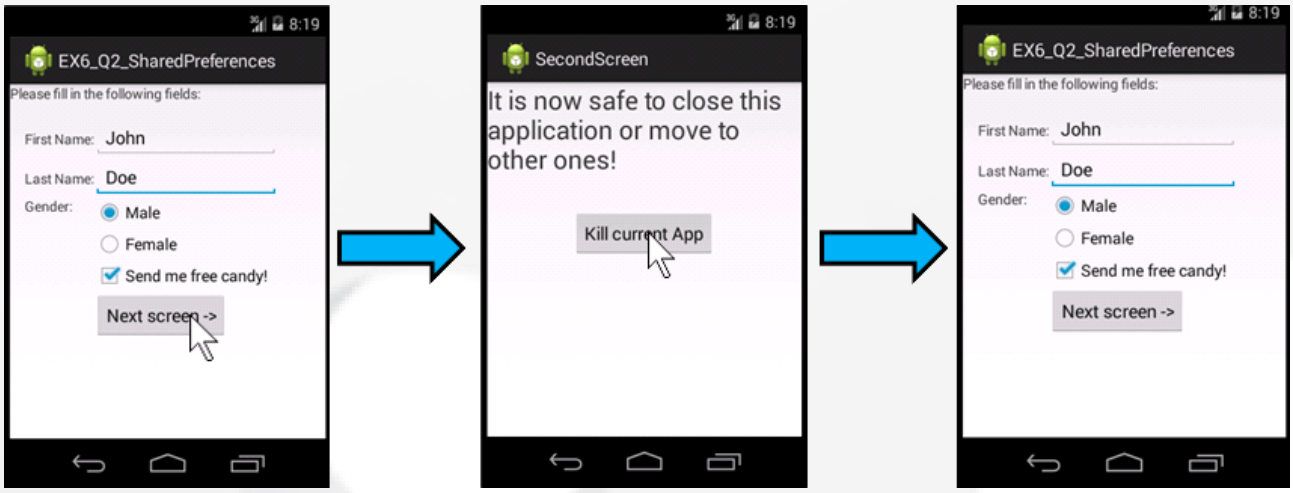
- First name
- Last name
- Gender (using a Radio control)
- A checkbox of your choosing
- Navigation button

Clicking the button navigates to a secondary screen, but upon leaving the initial screen all the values must be automatically stored, so if the application if forcedly closed when the secondary screen is visible, upon reopening the application – all the details inserted in the initial screen will be restored.

Tip: Use SharedPreferences to store field values.

Example:

Original state | **Navigate to 2nd screen** | **Re opening the app**



Menu

Question 3 –

Write an app that has a menu with the following options:

- Show a message – opens up a toast with a message from you
- Checkbox with the text of “awful color” – checking it will change the app background color into a god awful color. Un-checking this checkbox will return the original background color.
- Sub menu named “sub screens” which contains two options
  - Show about screen (opens an about screen activity)
  - Show Image screen (opens another activity that shows the user an image)

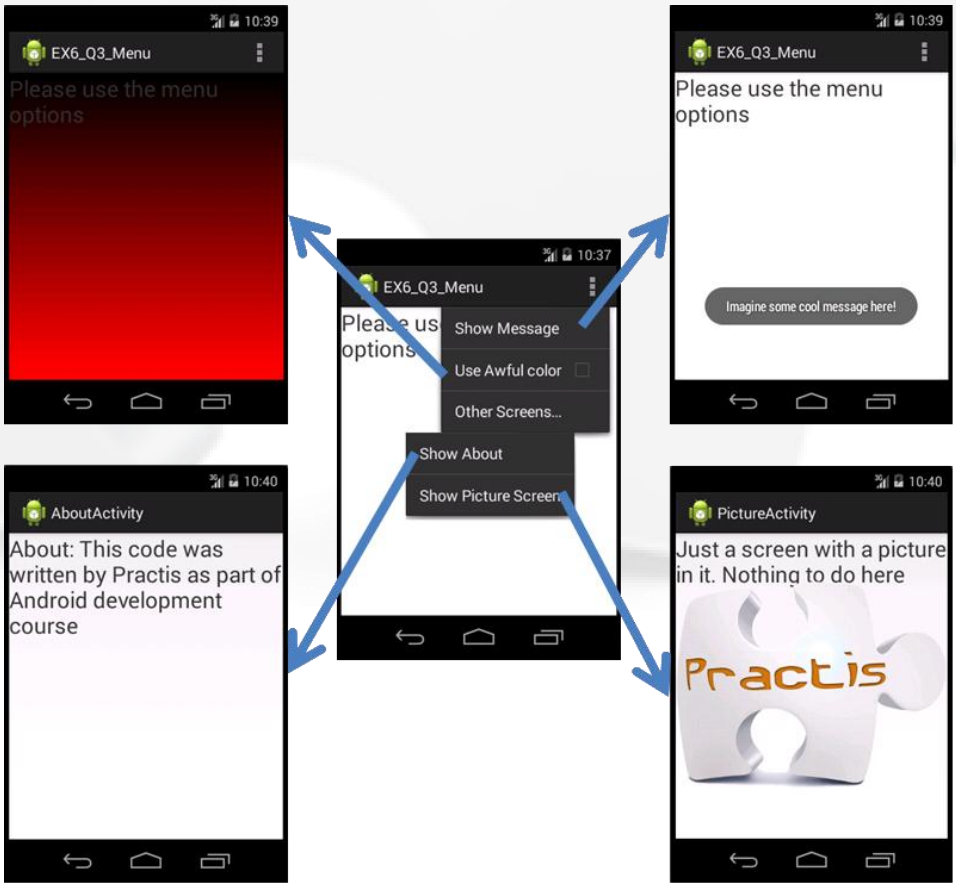
Tip: In order to mark a menu item as checkable it must be declared as such in the menu XML file:

```
android:checkable="true"
```

In addition, after it has been clicked you need to mark it as checked yourself (the Android OS won’t do it for us):

```
item.setChecked(true);
```

Example:



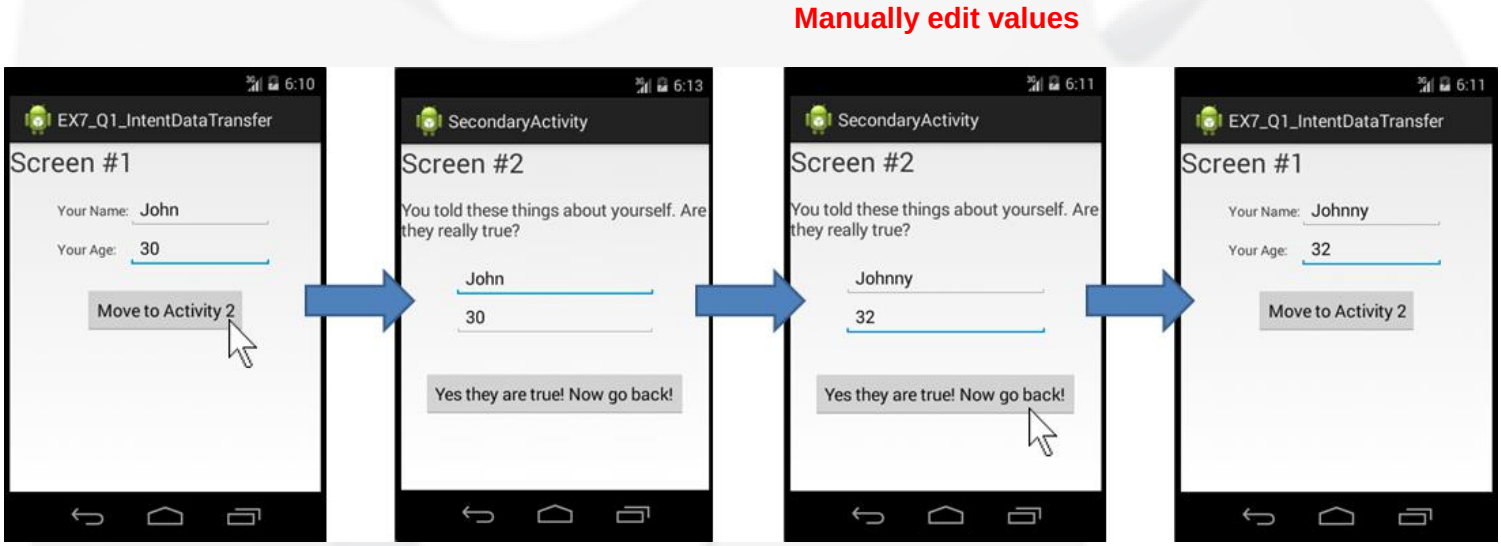
Intent

Question 4 –

This exercise is about data sharing between different activities. Write an app that has two different activities:

- The first one contains:
  - Several fields in which values need to be inserted
  - A button to navigate to the second screen
- The second one will show us:
  - The values provided in the first screen (but with a different styling).
  - These values are editable
  - A button that takes us back to the first screen. Clicking this button will update the original values of the first screen with their new value inserted in the second screen.

Example:



Question 5 –

This exercise is about using various intents.

Write an application that has several buttons, each performing a different task:

- Show the details of a random contact
- Open the camera and show the picture taken
- Open the browser to a pre-defined web page
- Open the SMS sending screen with preconfigured recipient and SMS content
- **Bonus:** Perform Google web search in a value provided in a EditText field (**Tip:** Use the ACTION_WEB_SEARCH)

Tip: To choose a random contact we need to use a randomization capability, which means we need to know how any contact are there. You can you this following code:

```
// get number of contacts so we can choose a random one

Cursor cursor =
getContentResolver().query(ContactsContract.Contacts.CONTENT_URI, null, null,
null, null);

int count = cursor.getCount() + 1;
```

You will need the following permission of this code to work:

```
    <uses-permission android:name="android.permission.READ_CONTACTS" />
```

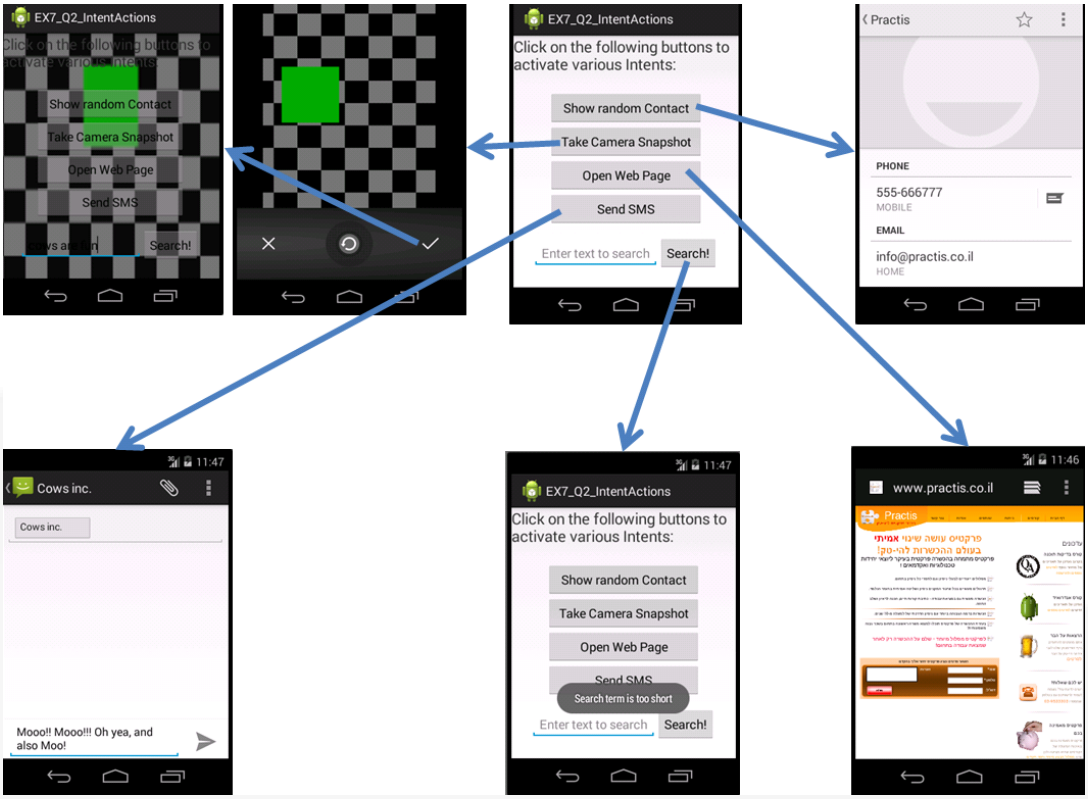
(Place this line in the AndroidManifest.xml file)

Now we can choose a random contact:

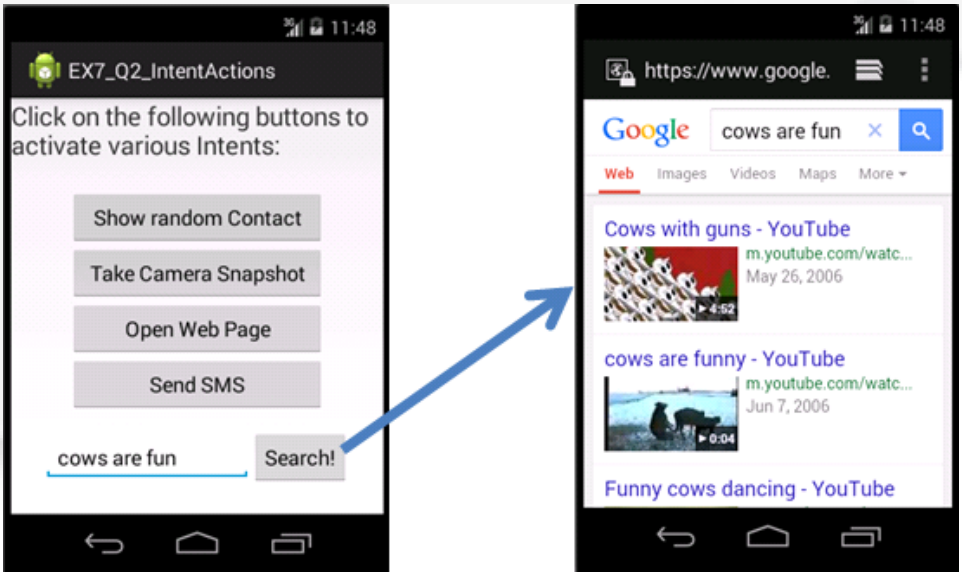
```
Random randomNumberGenerator = new Random();

int randomContact = randomNumberGenerator.nextInt(count);
```

Example:



Valid search example:



Good luck!