Mobile Monday Tech Lab 02

SOFTWARE CRAFTSMANSHIP GUILD

Android Development - Contact Book App

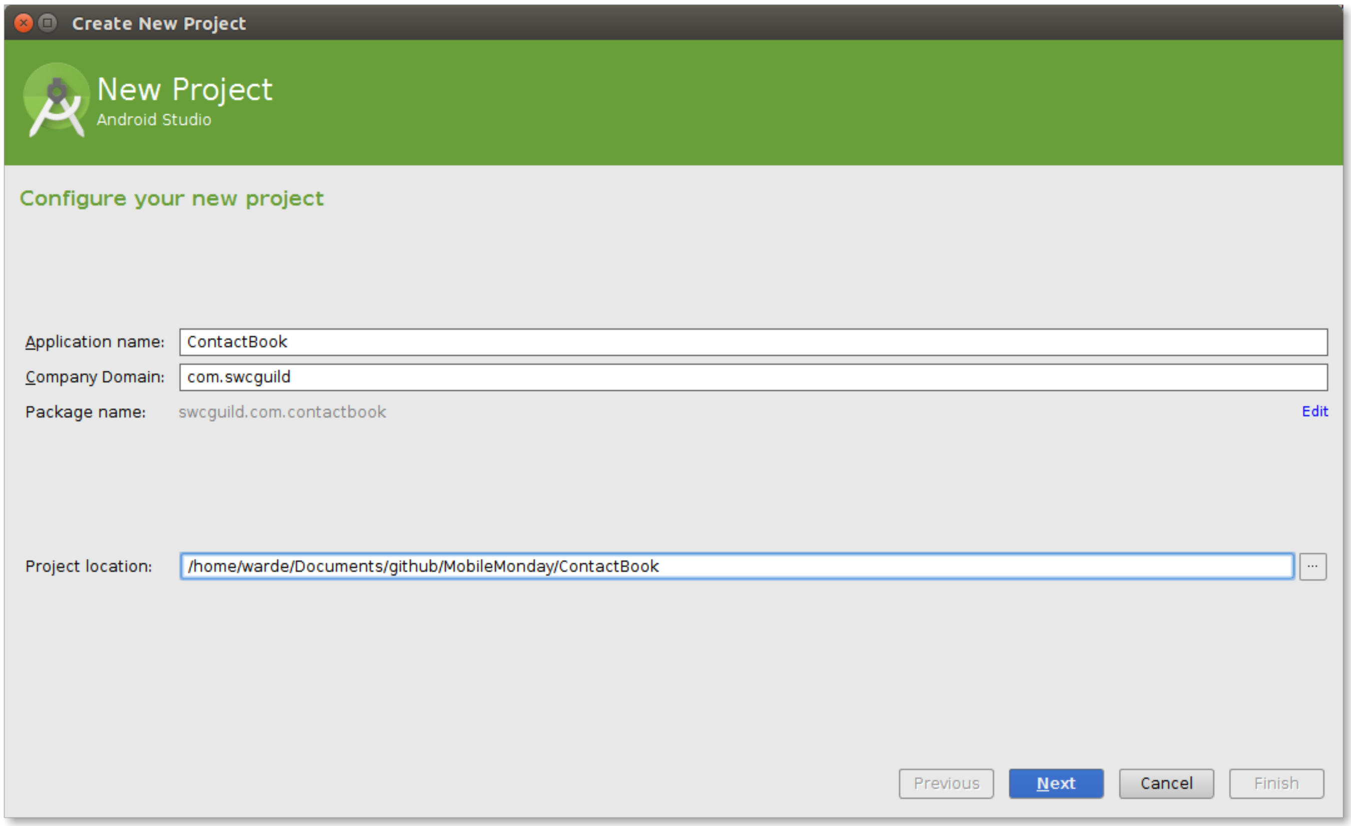
Overview

In this lab we are going to look at a more complicated Android app - a Contact manager. Due to the complexity of the application and time constraints on our session, we will not be coding the entire application live. Rather, we'll get the project started and then look overall design and major components of the completed project which can be found in GitHub @ https://github.com/swcguild/MobileMonday. Our objectives for this session are:

- Build a UI with Fragments and FragmentTransactions
- Understand how to communicate between Fragments and Main Activity using callbacks
- Understand how to save application state when phone configuration changes
- Define and apply styles to the UI
- Create and manipulate a SQLite database
- Create and use asynchronous tasks

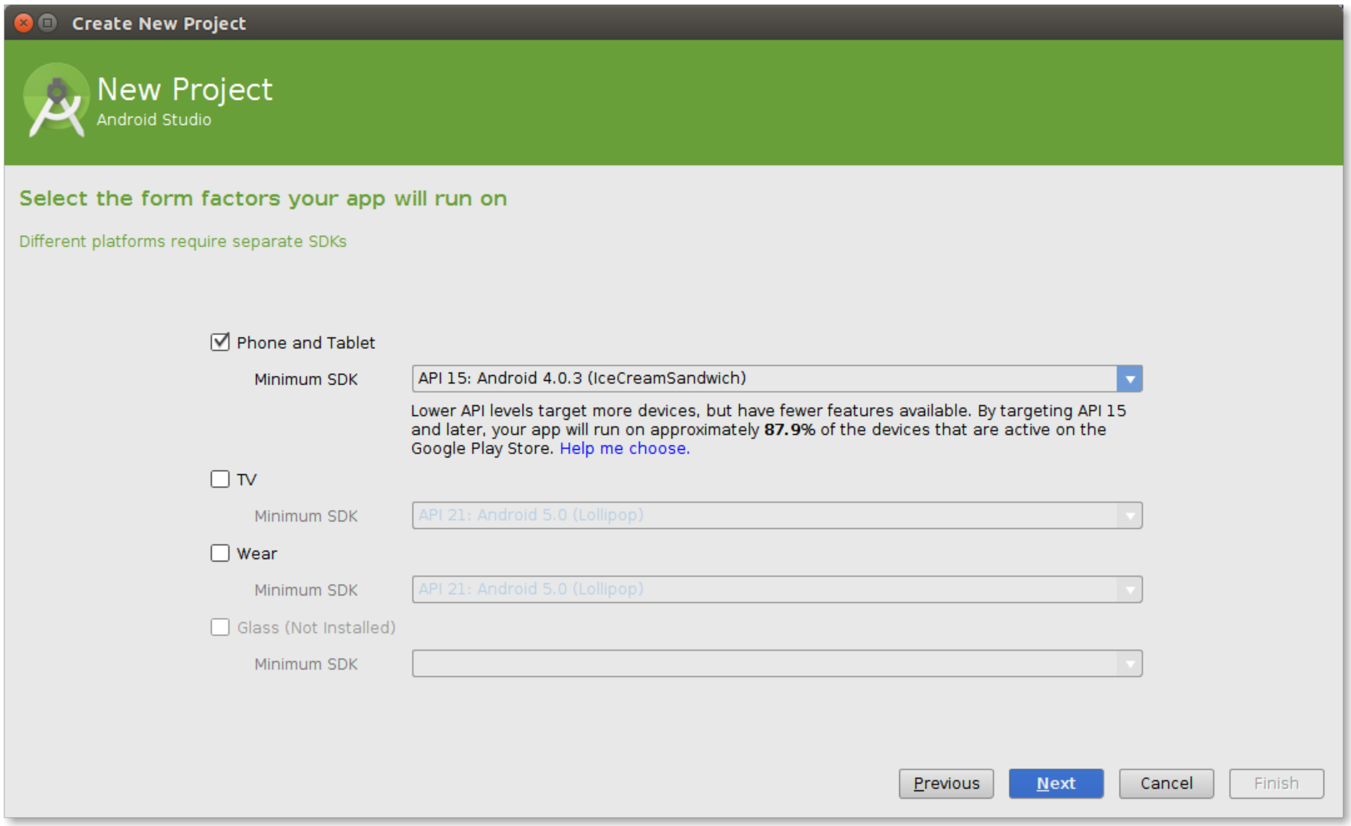
Creating the Project

This tutorial will step you through the initial creation of the project and the creation of the UI and Resource components. Begin by creating a new project called **ContactBook**. Match the values found below in the New Project Creation Wizard (a more detailed description of this process can be found in the notes from Tech Lab 01):



Application name: ContactBook
Company Domain: com.swcguild
Package name: swcguild.com.contactbook
Project location: /home/warde/Documents/github/MobileMonday/ContactBook

Initial Creation Wizard Screen



Select the form factors your app will run on

Different platforms require separate SDKs

[✓] Phone and Tablet
Minimum SDK: API 15: Android 4.0.3 (IceCreamSandwich)

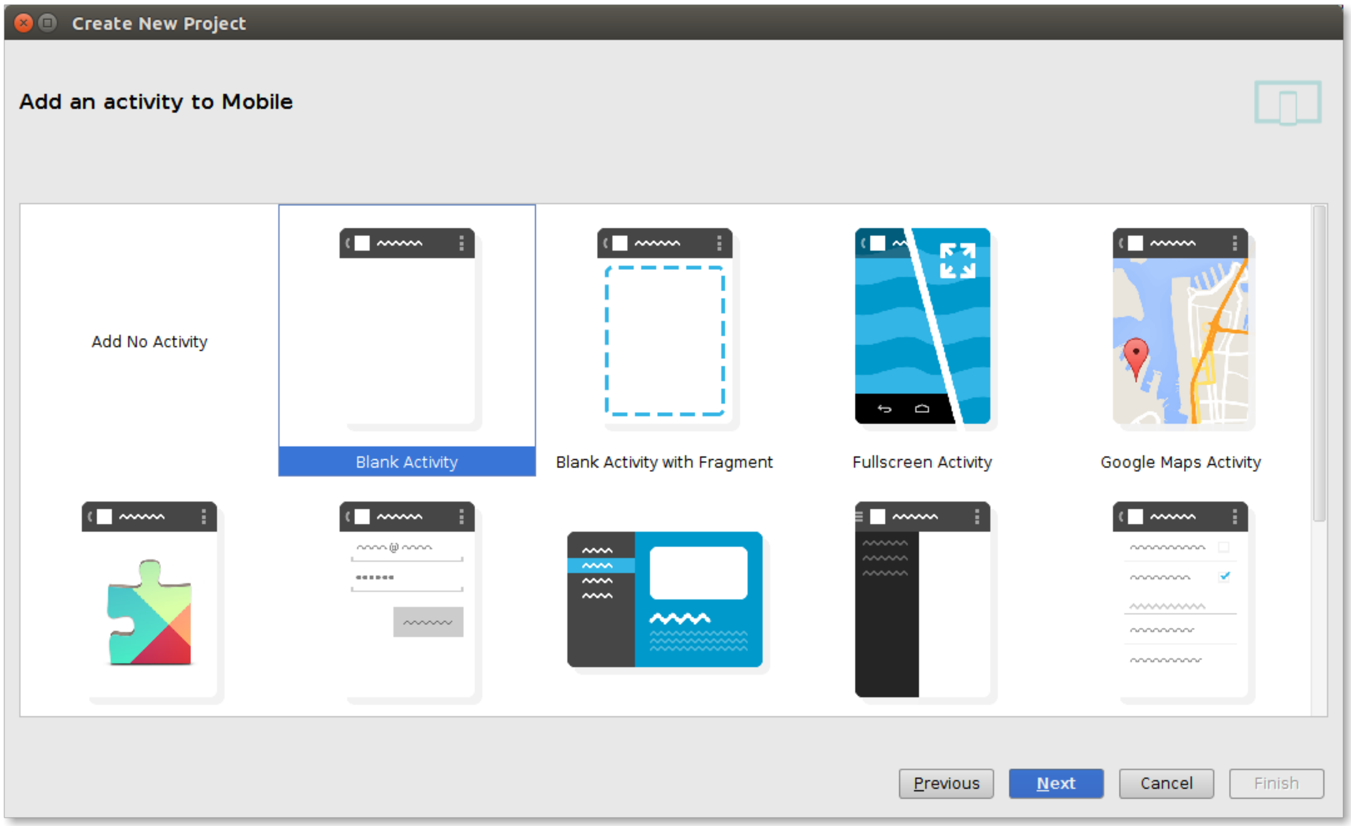
Lower API levels target more devices, but have fewer features available. By targeting API 15 and later, your app will run on approximately **87.9%** of the devices that are active on the Google Play Store. Help me choose.

[ ] TV
Minimum SDK: API 21: Android 5.0 (Lollipop)

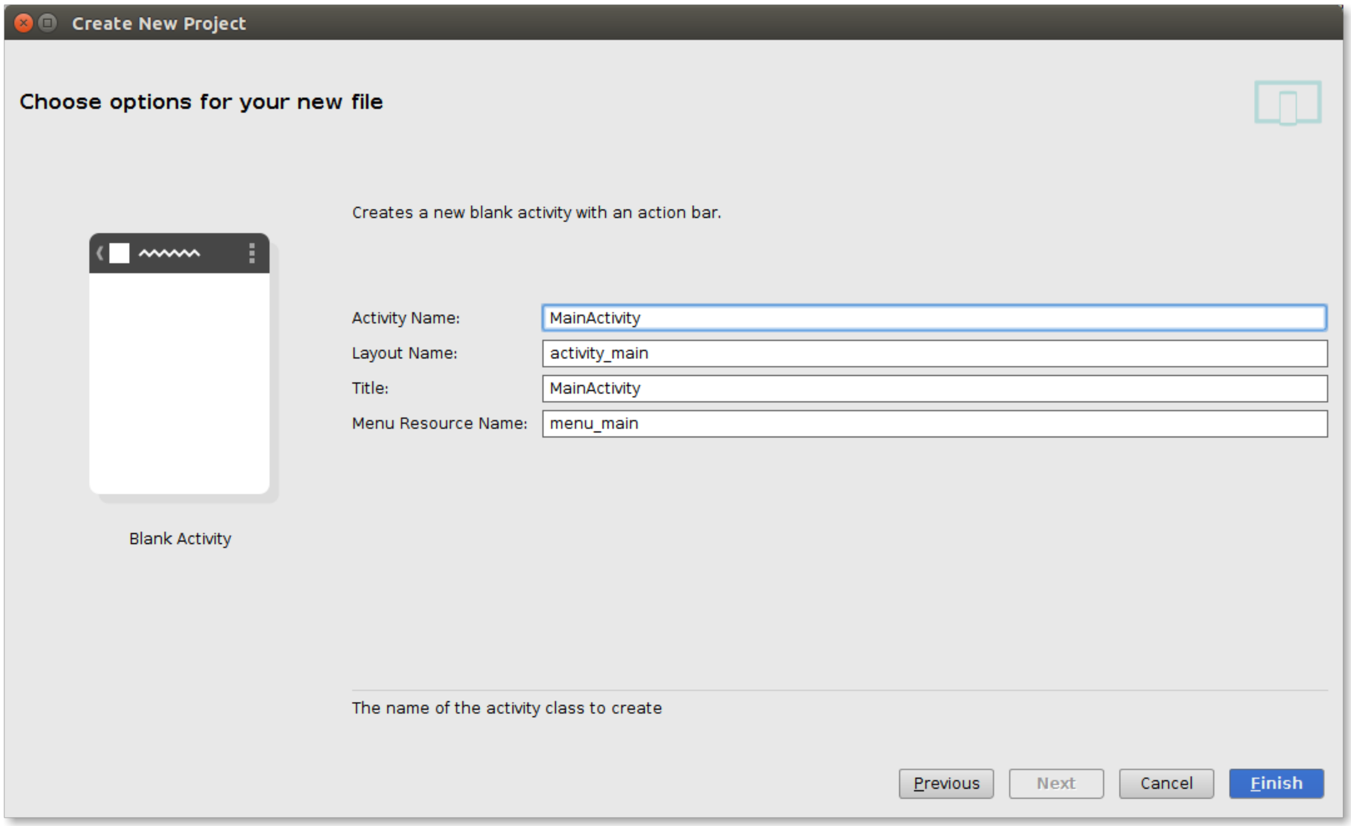
[ ] Wear
Minimum SDK: API 21: Android 5.0 (Lollipop)

[ ] Glass (Not Installed)
Minimum SDK:

Platform and SDK Options



Initial Activity Selection - Blank Activity Selected



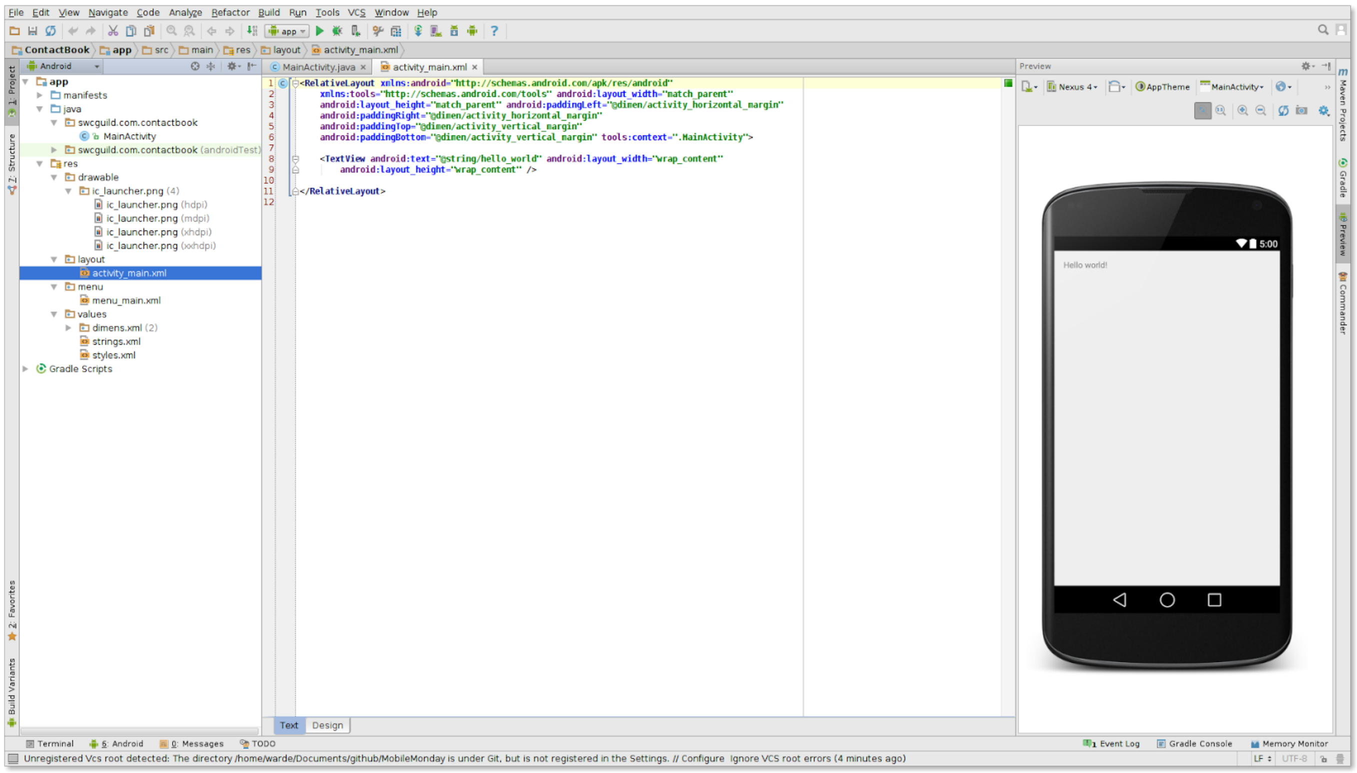
Choose options for your new file

Creates a new blank activity with an action bar.

Activity Name:	MainActivity
Layout Name:	activity_main
Title:	MainActivity
Menu Resource Name:	menu_main

The name of the activity class to create

MainActivity Options - Defaults Chosen



```
<RelativeLayout 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:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin" tools:context=".MainActivity">

    <TextView android:text="@string/hello_world" android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

```
Unregistered Vcs root detected: The directory /home/warde/Documents/github/MobileMonday is under Git, but is not registered in the Settings. // Configure  Ignore VCS root errors (4 minutes ago)
```

Newly Created Project

Creating the UI and Resource Files

In this section we'll walk through the creation of the UI layout and resource XML files. Android Studio has visual editors for most of these files but we will be looking directly at the XML in this document because it is easier to see all the settings and values. You can switch between XML Text and Designer mode in Android Studio and I encourage you to become familiar with both.

Strings

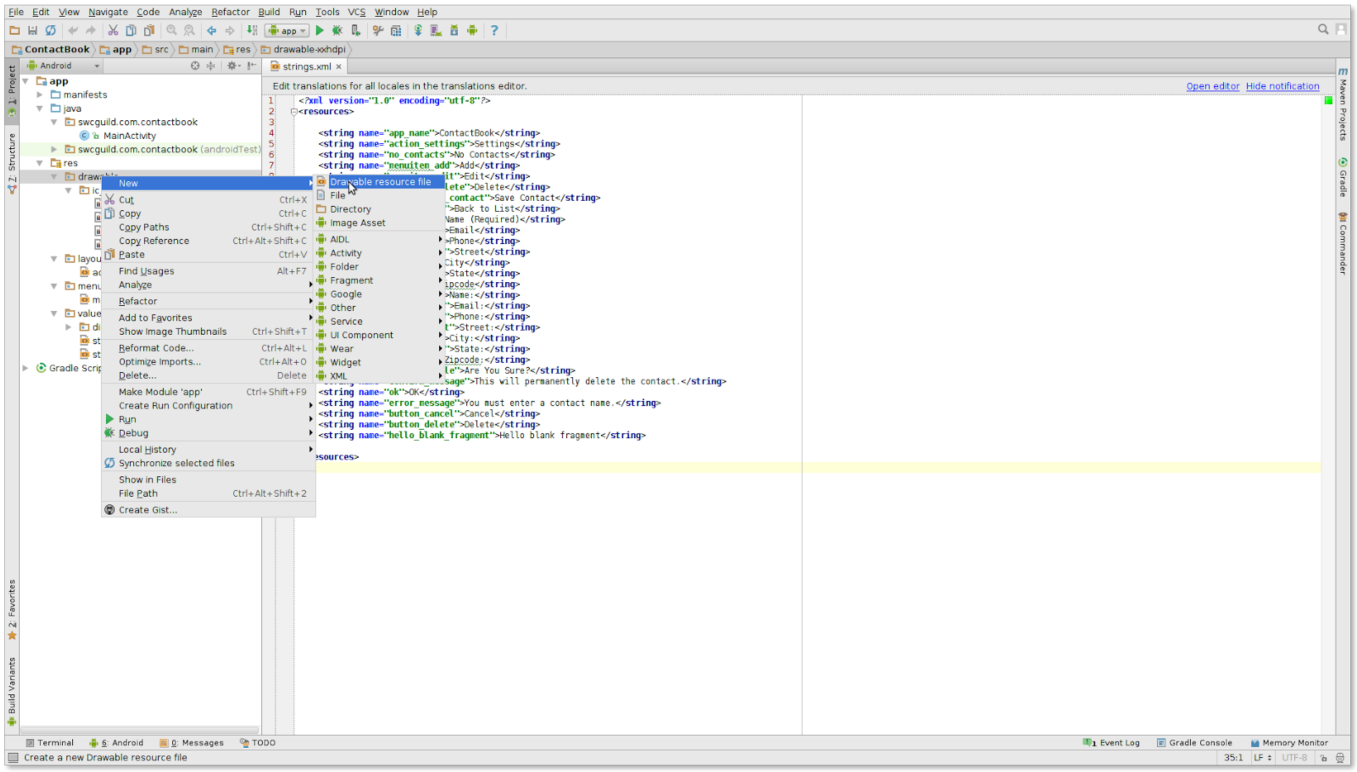
The file **strings.xml** contains String resources that can be used throughout our applications. The String values are used for things such as button text, dialog messages and text entry labels. It is considered good form to use String values from **string.xml** rather than hardcoded Strings. This is what your **strings.xml** should look like:

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">ContactBook</string>
    <string name="action_settings">Settings</string>
    <string name="no_contacts">No Contacts</string>
    <string name="menuitem_add">Add</string>
    <string name="menuitem_edit">Edit</string>
    <string name="menuitem_delete">Delete</string>
    <string name="button_save_contact">Save Contact</string>
    <string name="button_back">Back to List</string>
    <string name="hint_name">Name (Required)</string>
    <string name="hint_email">Email</string>
    <string name="hint_phone">Phone</string>
    <string name="hint_street">Street</string>
    <string name="hint_city">City</string>
    <string name="hint_state">State</string>
    <string name="hint_zip">Zipcode</string>
    <string name="label_name">Name:</string>
    <string name="label_email">Email:</string>
    <string name="label_phone">Phone:</string>
    <string name="label_street">Street:</string>
    <string name="label_city">City:</string>
    <string name="label_state">State:</string>
    <string name="label_zip">Zipcode:</string>
    <string name="confirm_title">Are You Sure?</string>
    <string name="confirm_message">This will permanently delete the contact.</string>
    <string name="ok">OK</string>
    <string name="error_message">You must enter a contact name.</string>
    <string name="button_cancel">Cancel</string>
    <string name="button_delete">Delete</string>
    <string name="hello_blank_fragment">Hello blank fragment</string>
</resources>
```

Borders

We'll now create a file called **textview_border.xml** in the **drawable** folder:

- Right click on 'drawable' and select New Drawable Resource

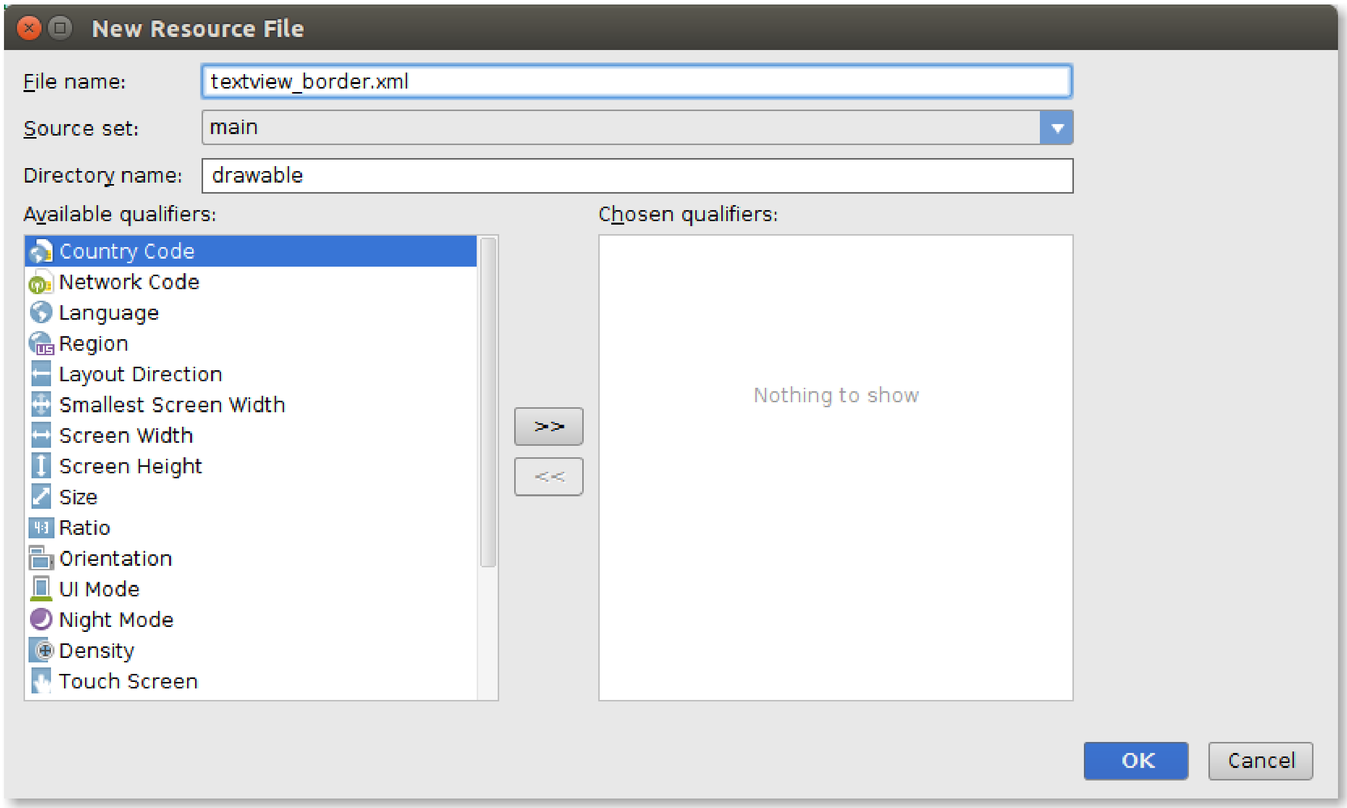


```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">ContactBook</string>
    <string name="action_settings">Settings</string>
    <string name="no_contacts">No Contacts</string>
    <string name="menuitem_add">Add</string>
```

```
    <string name="ok">OK</string>
    <string name="error_message">You must enter a contact name.</string>
    <string name="button_cancel">Cancel</string>
    <string name="button_delete">Delete</string>
    <string name="hello_blank_fragment">Hello blank fragment</string>
```

New Drawable Resource Selection

Fill out the New Resource File dialog with the following values and click OK:


New Resource File

File name: textview_border.xml

Source set: main

Directory name: drawable

Available qualifiers:

- Country Code
- Network Code
- Language
- Region
- Layout Direction
- Smallest Screen Width
- Screen Width
- Screen Height
- Size
- Ratio
- Orientation
- UI Mode
- Night Mode
- Density
- Touch Screen

Chosen qualifiers:

Nothing to show

New Resource File Dialog

Put the following XML into your newly created file. This file defines a rectangle with rounded corners:

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <corners android:radius="5dp" />
    <stroke
        android:width="1dp"
        android:color="#555" />
    <padding
        android:bottom="10dp"
        android:left="10dp"
        android:right="10dp"
        android:top="10dp" />
</shape>
```

Android text input fields do not have borders by default - this definition allows us to apply a border to all of these fields.

Styles

Application styles are defined in **styles.xml**. We will define the overall theme for our application as well as styles for Labels and TextView input components. Replace the contents of **styles.xml** with the following:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
    </style>

    <style name="ContactLabelTextView">
        <item name="android:layout_width">wrap_content</item>
        <item name="android:layout_height">wrap_content</item>
        <item name="android:layout_gravity">right|center_vertical</item>
    </style>

    <style name="ContactTextView">
        <item name="android:layout_width">wrap_content</item>
        <item name="android:layout_height">wrap_content</item>
        <item name="android:layout_gravity">fill_horizontal</item>
        <item name="android:textSize">16sp</item>
        <item name="android:background">@drawable/textview_border</item>
    </style>
</resources>
```

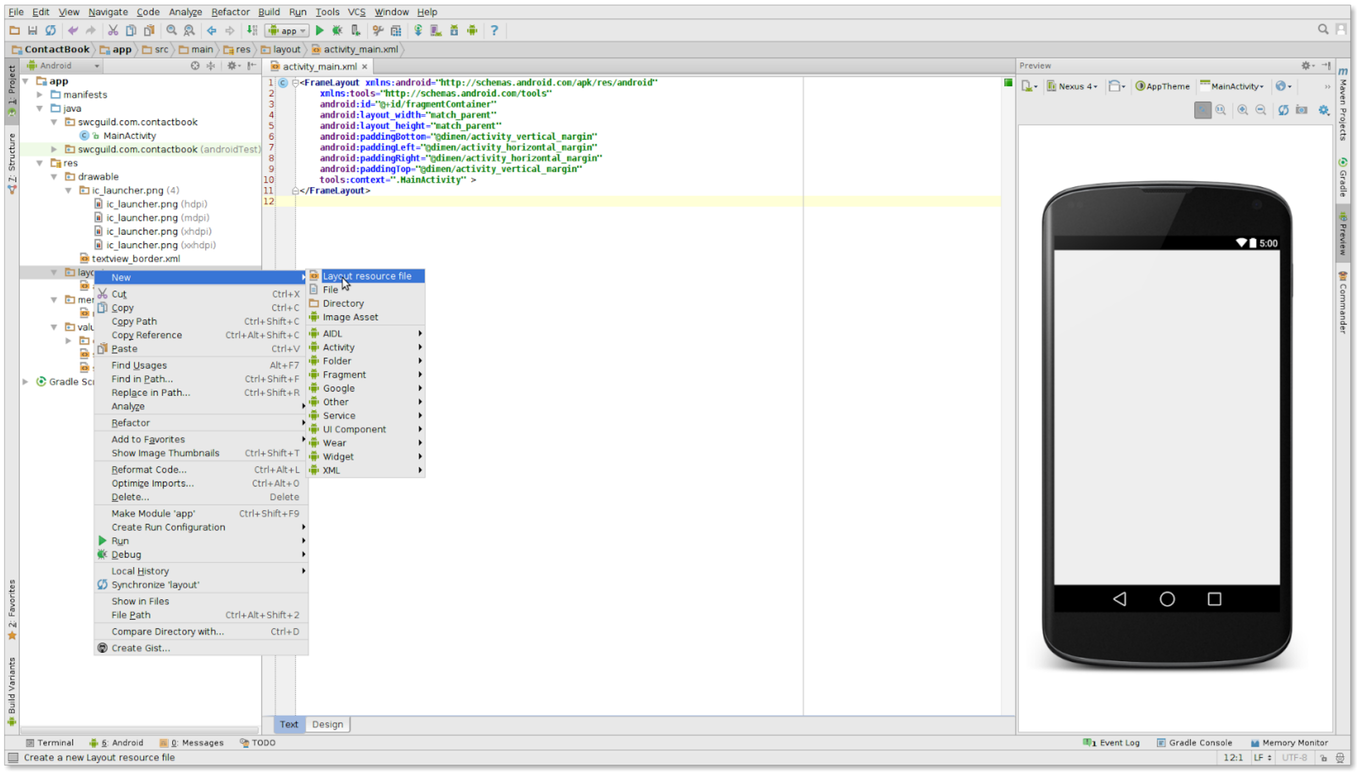
Main Activity Layout

The file **activity_main.xml** defines the layout for the component that will act as the container for all of our layout fragments. Replace the contents of your **activity_main.xml** file with the following:

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/fragmentContainer"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >
</FrameLayout>
```

UI Layout Files

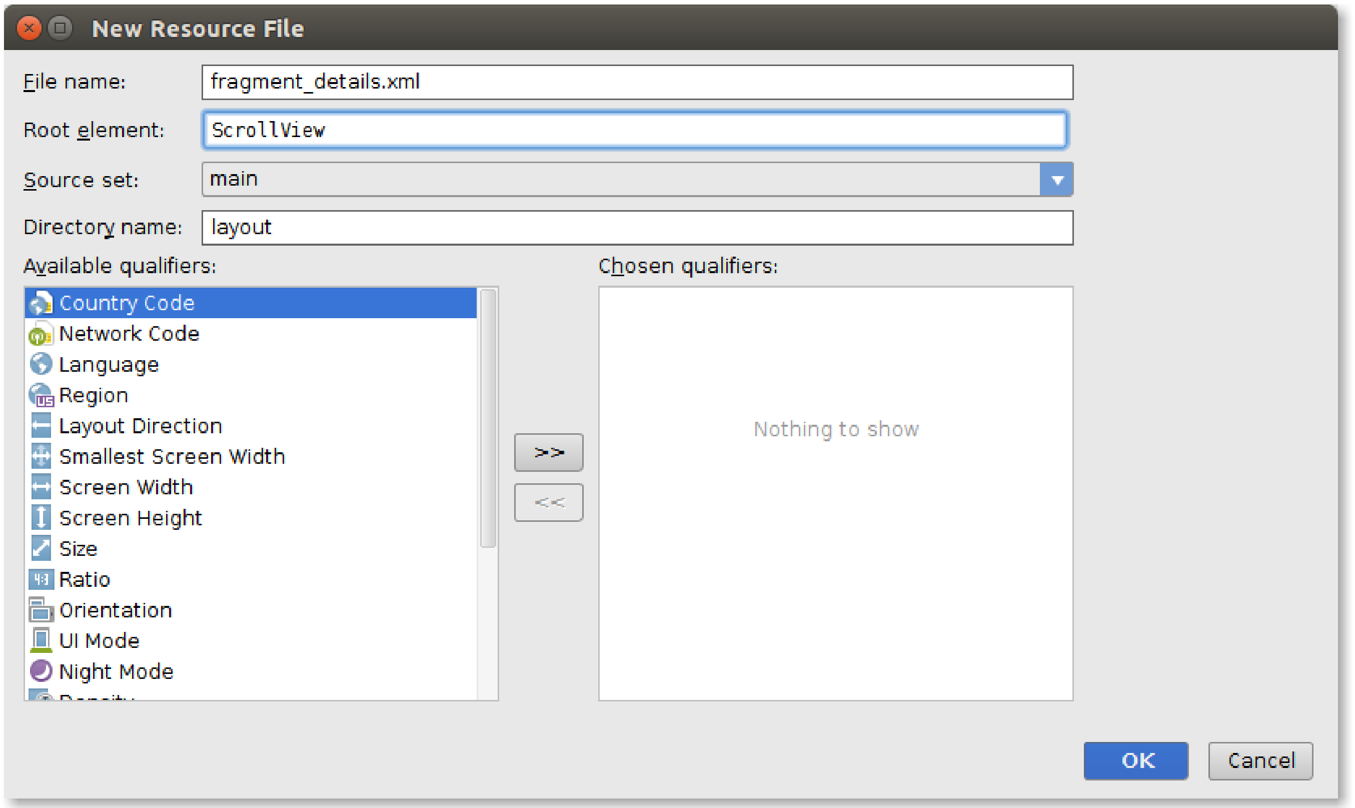
The next two XML files define the layout for our Contact Details screen and Add/Edit Contact screen respectively. You must create these files manually by doing the following :



```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/fragmentContainer"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >
</FrameLayout>
```

New Layout Resource File

DetailsFragment Layout


New Resource File

File name: fragment_details.xml

Root element: ScrollView

Source set: main

Directory name: layout

Available qualifiers:

- Country Code
- Network Code
- Language
- Region
- Layout Direction
- Smallest Screen Width
- Screen Width
- Screen Height
- Size
- Ratio
- Orientation
- UI Mode
- Night Mode

Chosen qualifiers:

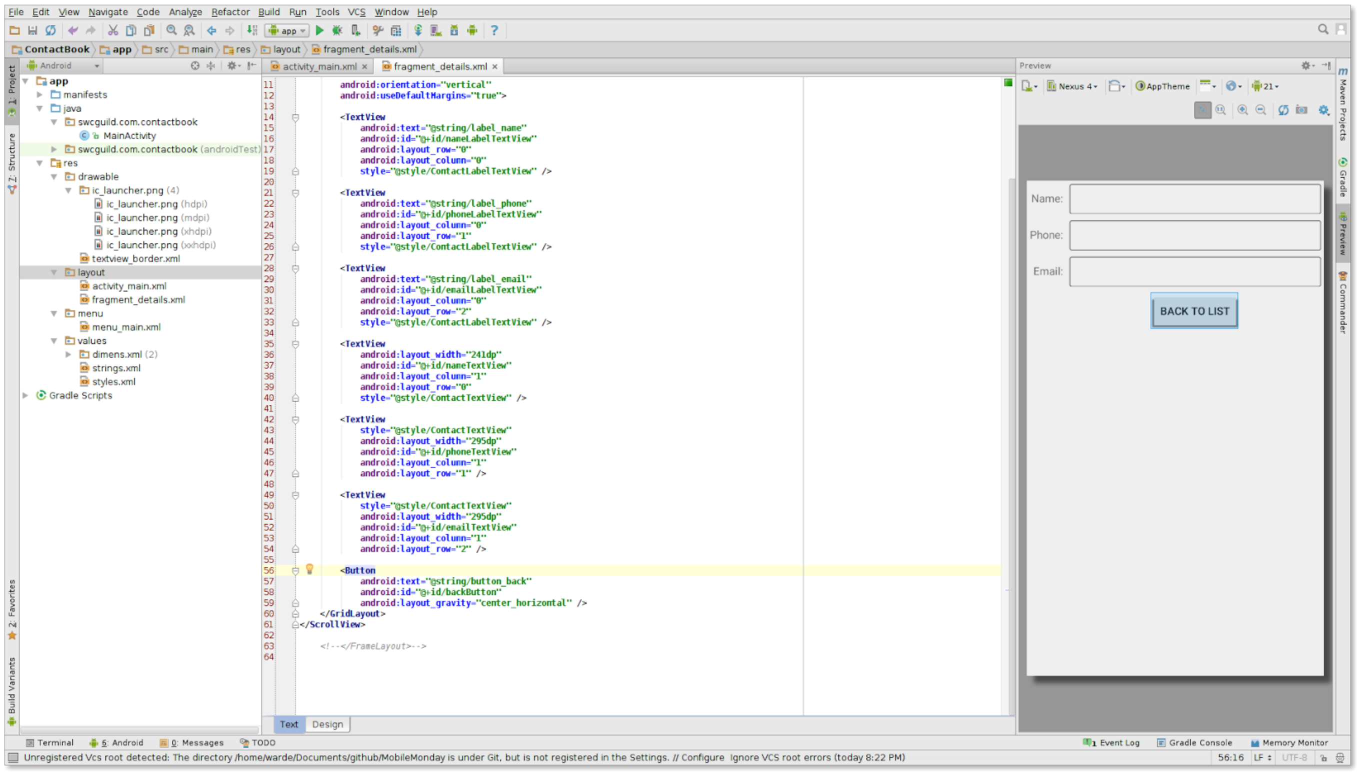
Nothing to show

New Layout Resource Dialog - fragment_details.xml

The file **fragment_details.xml** defines the layout of the screen that displays the details of a contact. The main element in this layout is a ScrollView component. Inside the ScrollView is a GridLayout (1 column) that contains display components for the contact fields and a button that allows the user to go back to the List Contacts screen. After creating **fragment_details.xml**, replace the contents with the following:

```xml
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/detailsScrollView" >

    <GridLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:columnCount="2"
        android:orientation="vertical"
        android:useDefaultMargins="true">

        <TextView
            android:text="@string/label_name"
            android:id="@+id/nameLabelTextView"
            android:layout_row="0"
            android:layout_column="0"
            style="@style/ContactLabelTextView" />

        <TextView
            android:text="@string/label_phone"
            android:id="@+id/phoneLabelTextView"
            android:layout_column="0"
            android:layout_row="1"
            style="@style/ContactLabelTextView" />

        <TextView
            android:text="@string/label_email"
            android:id="@+id/emailLabelTextView"
            android:layout_column="0"
            android:layout_row="2"
            style="@style/ContactLabelTextView" />

        <TextView
            android:layout_width="241dp"
            android:id="@+id/nameTextView"
            android:layout_column="1"
            android:layout_row="0"
            style="@style/ContactTextView" />
        <TextView
            style="@style/ContactTextView"
            android:layout_width="295dp"
            android:id="@+id/phoneTextView"
            android:layout_column="1"
            android:layout_row="1" />

        <TextView
            style="@style/ContactTextView"
            android:layout_width="295dp"
            android:id="@+id/emailTextView"
            android:layout_column="1"
            android:layout_row="2" />

        <Button
            android:text="@string/button_back"
            android:id="@+id/backButton"
            android:layout_gravity="center_horizontal" />
    </GridLayout>
</ScrollView>
```

Android Studio should display the following for **fragment_details.xml**:



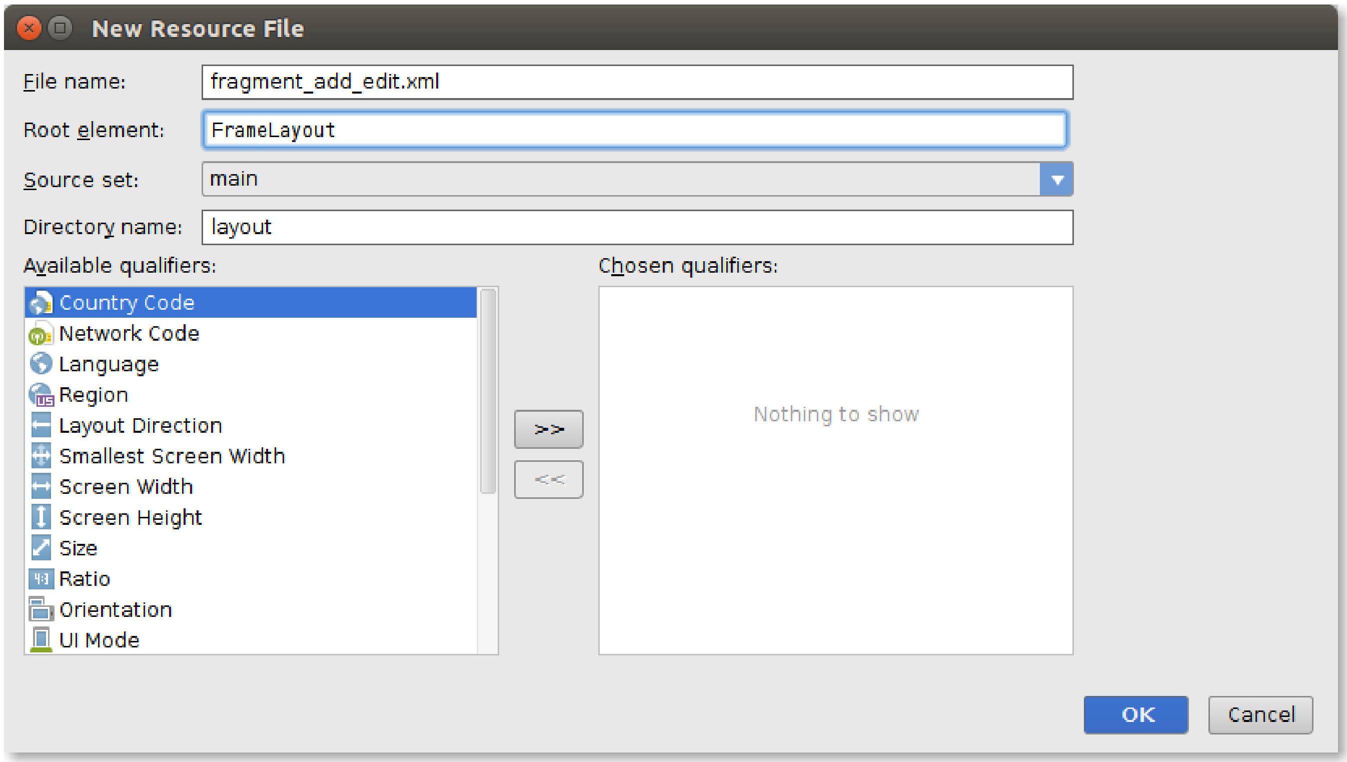
```
        android:orientation="vertical"
        android:useDefaultMargins="true">

        <TextView
            android:text="@string/label_name"
            android:id="@+id/nameLabelTextView"
            android:layout_row="0"
            android:layout_column="0"
            style="@style/ContactLabelTextView" />

        <TextView
            android:text="@string/label_phone"
            android:id="@+id/phoneLabelTextView"
            android:layout_column="0"
            android:layout_row="1"
            style="@style/ContactLabelTextView" />

        <TextView
            android:text="@string/label_email"
            android:id="@+id/emailLabelTextView"
            android:layout_column="0"
            android:layout_row="2"
            style="@style/ContactLabelTextView" />

        <TextView
            android:layout_width="241dp"
            android:id="@+id/nameTextView"
            android:layout_column="1"
            android:layout_row="0"
            style="@style/ContactTextView" />

        <TextView
            style="@style/ContactTextView"
            android:layout_width="295dp"
            android:id="@+id/phoneTextView"
            android:layout_column="1"
            android:layout_row="1" />

        <TextView
            style="@style/ContactTextView"
            android:layout_width="295dp"
            android:id="@+id/emailTextView"
            android:layout_column="1"
            android:layout_row="2" />

        <Button
            android:text="@string/button_back"
            android:id="@+id/backButton"
            android:layout_gravity="center_horizontal" />
    </GridLayout>
</ScrollView>

    <!--</FrameLayout>-->
```

Unregistered Vcs root detected: The directory /home/warde/Documents/github/MobileMonday is under Git, but is not registered in the Settings. // Configure Ignore VCS root errors (today 8:22 PM)

fragment_details.xml

AddEditFragment Layout

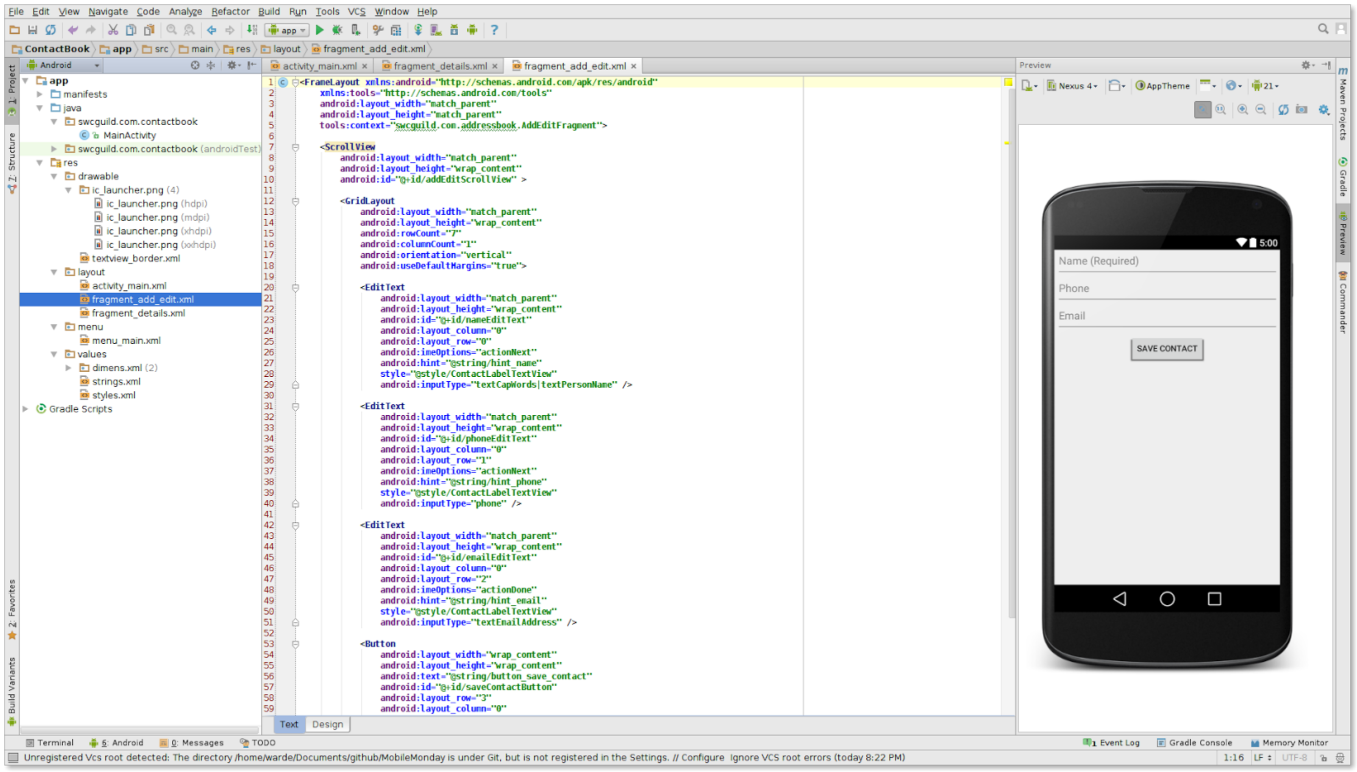
The file **fragment_add_edit.xml** defines the layout of the screen that displays the form for adding and editing contacts. The main element in this layout is a FrameLayout containing a ScrollView component. Inside the ScrollView is a GridLayout (1 column) that contains display components for the contact fields and a button that allows the user to save the contact. After creating **fragment_add_edit.xml**, replace the contents with the following:



File name: fragment_add_edit.xml
Root element: FrameLayout
Source set: main
Directory name: layout

New Layout Resource File Dialog - fragment_add_edit.xml

```
<FrameLayout 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="swcguild.com.addressbook.AddEditFragment">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/addEditScrollView" >

        <GridLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:rowCount="7"
            android:columnCount="1"
            android:orientation="vertical"
            android:useDefaultMargins="true">

            <EditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:id="@+id/nameEditText"
                android:layout_column="0"
                android:layout_row="0"
                android:imeOptions="actionNext"
                android:hint="@string/hint_name"
                style="@style/ContactLabelTextView"
                android:inputType="textCapWords|textPersonName" />

            <EditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:id="@+id/phoneEditText"
                android:layout_column="0"
                android:layout_row="1"
                android:imeOptions="actionNext"
                android:hint="@string/hint_phone"
                style="@style/ContactLabelTextView"
                android:inputType="phone" />

            <EditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:id="@+id/emailEditText"
                android:layout_column="0"
                android:layout_row="2"
                android:imeOptions="actionDone"
                android:hint="@string/hint_email"
                style="@style/ContactLabelTextView"
                android:inputType="textEmailAddress" />

            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/button_save_contact"
                android:id="@+id/saveContactButton"
                android:layout_row="3"
                android:layout_column="0"
                android:layout_gravity="center_horizontal" />
        </GridLayout>
    </ScrollView>

</FrameLayout>
```



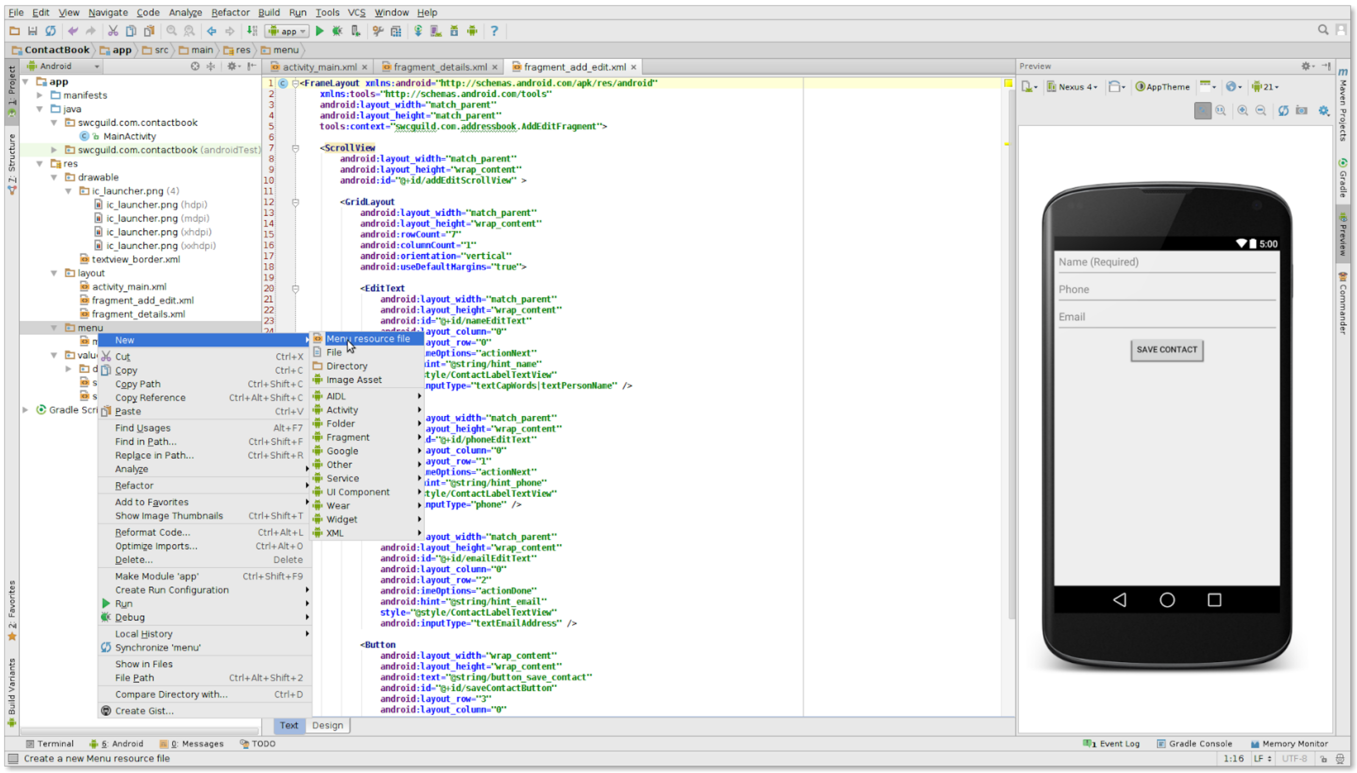
```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="swcguild.com.addressbook.AddEditFragment">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/addEditScrollView" >

        <GridLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:rowCount="7"
            android:columnCount="1"
            android:orientation="vertical"
            android:useDefaultMargins="true">

            <EditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:id="@+id/nameEditText"
                android:layout_column="0"
                android:layout_row="0"
                android:imeOptions="actionNext"
                android:hint="@string/hint_name"
                style="@style/ContactLabelTextView"
                android:inputType="textCapWords|textPersonName" />

            <EditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:id="@+id/phoneEditText"
                android:layout_column="0"
                android:layout_row="1"
                android:imeOptions="actionNext"
                android:hint="@string/hint_phone"
                style="@style/ContactLabelTextView"
                android:inputType="phone" />

            <EditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:id="@+id/emailEditText"
                android:layout_column="0"
                android:layout_row="2"
                android:imeOptions="actionDone"
                android:hint="@string/hint_email"
                style="@style/ContactLabelTextView"
                android:inputType="textEmailAddress" />

            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/button_save_contact"
                android:id="@+id/saveContactButton"
                android:layout_row="3"
                android:layout_column="0"
```

```
Unregistered Vcs root detected: The directory /home/warde/Documents/github/MobileMonday is under Git, but is not registered in the Settings. // Configure  Ignore VCS root errors (today 8:22 PM)
```

fragment_add_edit.xml

Option Menu Items

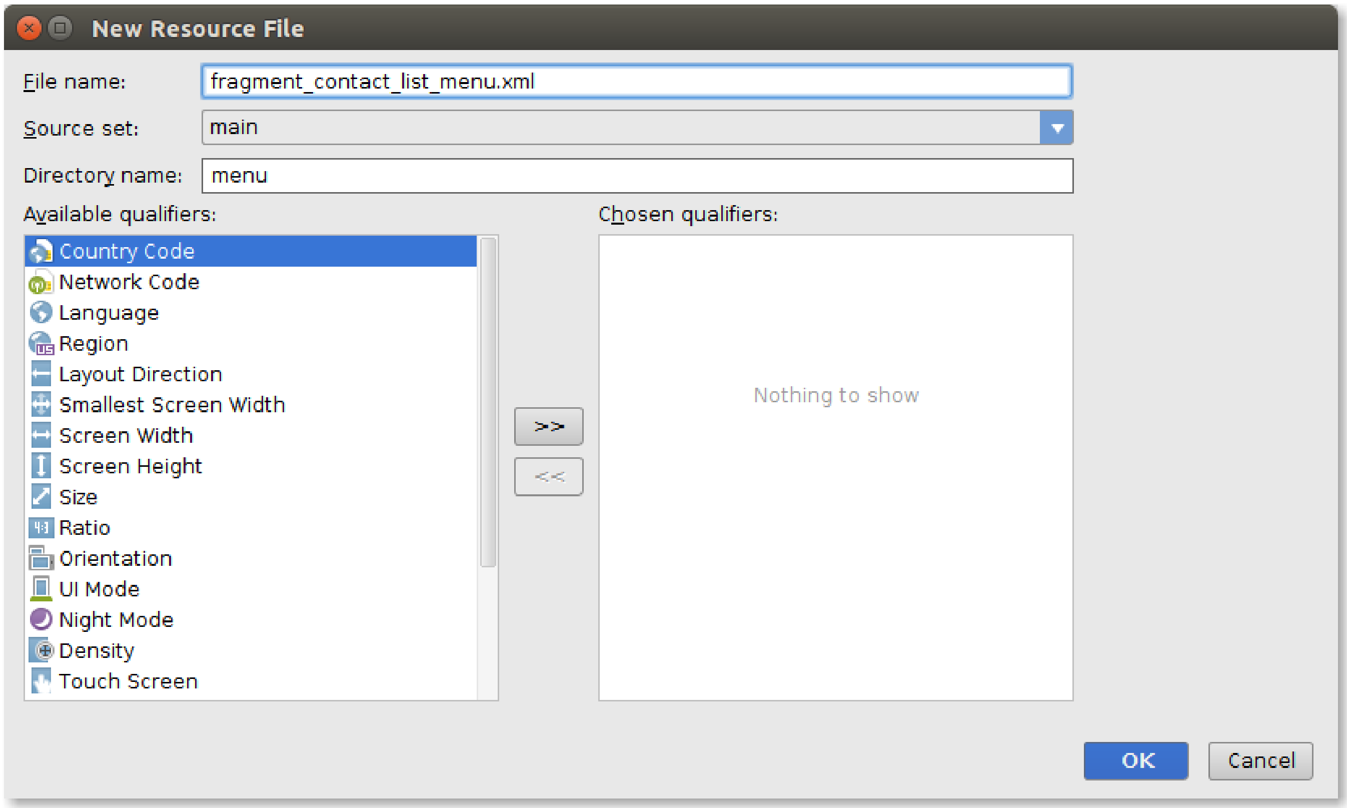
The next two files define the Add and Edit/Delete option menus. These option menus create buttons that, when clicked, allow the user to add, edit or delete a contact depending on which button is clicked. You must create these file manually (much as you did for the layout files above) by doing the following:



```
<FrameLayout 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="swcguild.com.addressbook.AddEditFragment">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/addEditScrollView" >

        <GridLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:rowCount="7"
            android:columnCount="1"
            android:orientation="vertical"
            android:useDefaultMargins="true">

            <EditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:id="@+id/nameEditText"
                      ayout_column="0"
                      ayout_row="0"
                      meOptions="actionNext"
                      int="@string/hint_name"
                      tyle/ContactLabelTextView"
                      nputType="textCapWords|textPersonName" />

                      ayout_width="match_parent"
                      ayout_height="wrap_content"
                      d="@+id/phoneEditText"
                      ayout_column="0"
                      ayout_row="1"
                      meOptions="actionNext"
                      int="@string/hint_phone"
                      tyle/ContactLabelTextView"
                      nputType="phone" />

                      ayout_width="match_parent"
                android:layout_height="wrap_content"
                android:id="@+id/emailEditText"
                android:layout_column="0"
                android:layout_row="2"
                android:imeOptions="actionDone"
                android:hint="@string/hint_email"
                style="@style/ContactLabelTextView"
                android:inputType="textEmailAddress" />

            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/button_save_contact"
                android:id="@+id/saveContactButton"
                android:layout_row="3"
                android:layout_column="0"
```

New Menu Resource File

Add Contact Menu


New Resource File

File name: fragment_contact_list_menu.xml

Source set: main

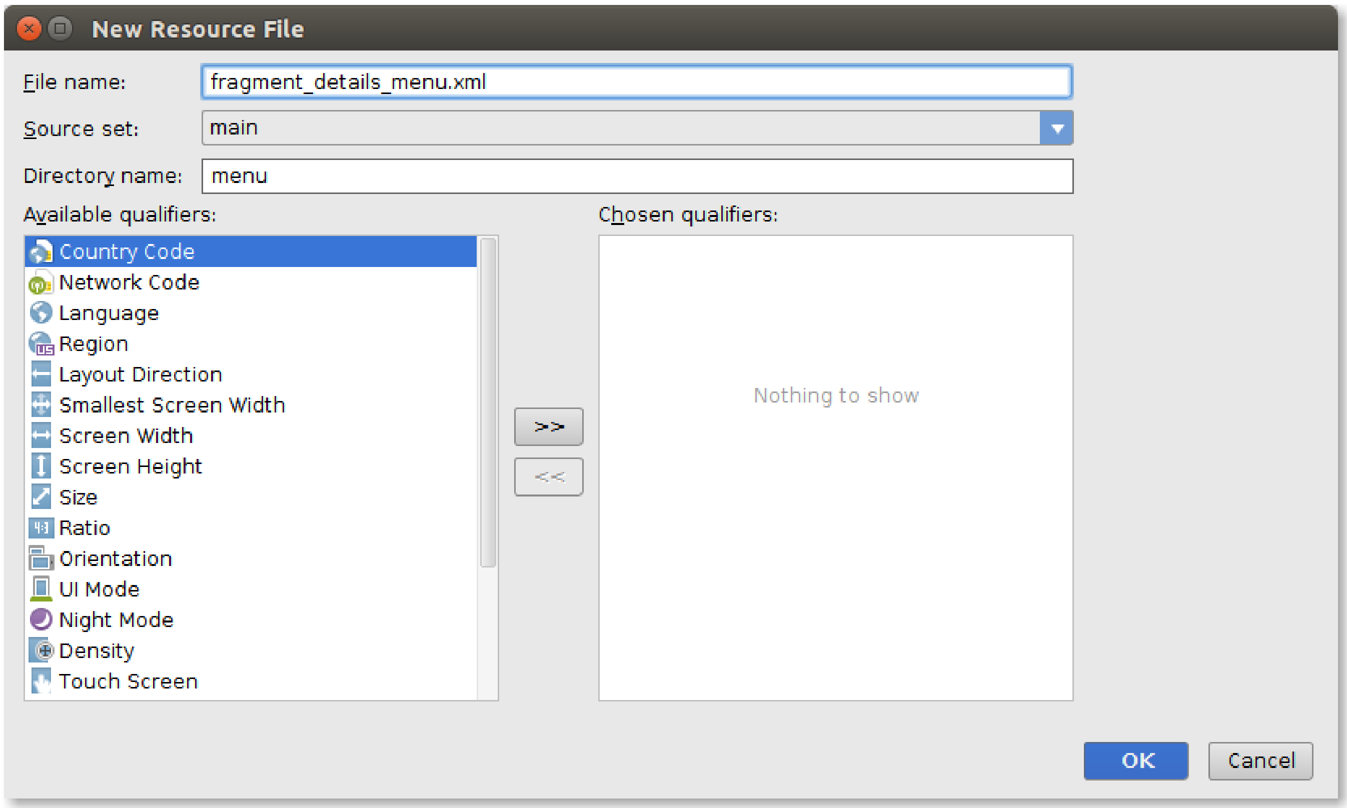
Directory name: menu

New Menu Resource File Dialog - fragment_contact_list_menu.xml

After you have created **fragment_contact_list_menu.xml**, replace the contents with the following (this file defines the Add option menu item):

```
<?xml version="1.0" encoding="utf-8"?>
<menu
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/action_add"
        android:orderInCategory="0"
        android:title="@string/menuitem_add"
        android:icon="@android:drawable/ic_menu_add"
        app:showAsAction="ifRoom|withText"
     />
</menu>
```

Edit/Delete Menu



New Resource File

File name: fragment_details_menu.xml

Source set: main

Directory name: menu

Available qualifiers: Country Code, Network Code, Language, Region, Layout Direction, Smallest Screen Width, Screen Width, Screen Height, Size, Ratio, Orientation, UI Mode, Night Mode, Density, Touch Screen

Chosen qualifiers: Nothing to show

New Menu Resource File Dialog - fragment_details_menu.xml

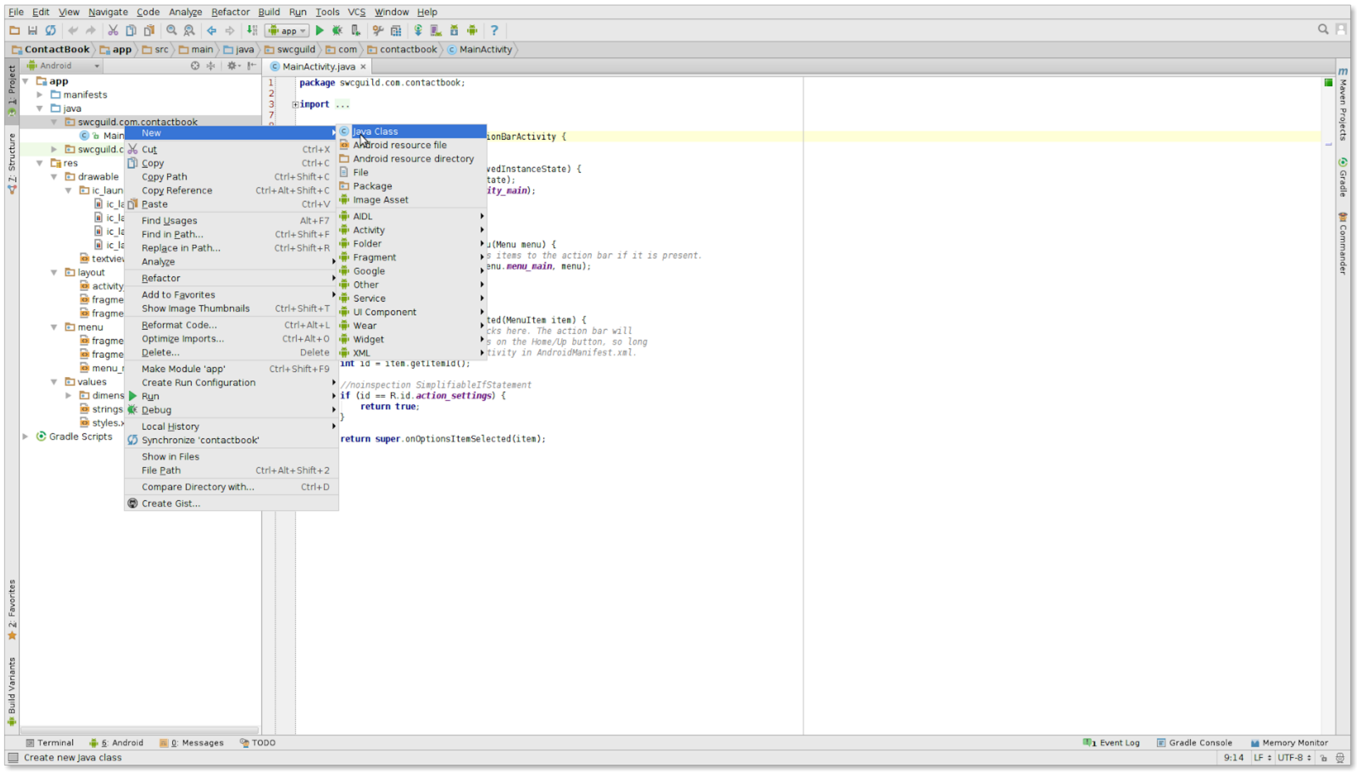
After you have created **fragment_details_menu.xml**, replace the contents with the following (this file defines the Edit and Delete option menu items):

```
<?xml version="1.0" encoding="utf-8"?>
<menu
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <item
        android:id="@+id/action_edit"
        android:orderInCategory="1"
        android:title="@string/menuitem_edit"
        android:icon="@android:drawable/ic_menu_edit"
        app:showAsAction="ifRoom|withText"
        />

    <item
        android:id="@+id/action_delete"
        android:orderInCategory="2"
        android:title="@string/menuitem_delete"
        android:icon="@android:drawable/ic_menu_delete"
        app:showAsAction="ifRoom|withText"
        />

</menu>
```

Creating the Java Classes

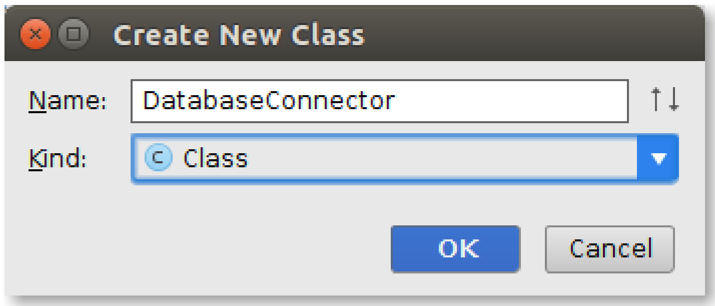
Now we will turn our attention to the Java class of the app. We will just point out the main features of each class - more detailed information can be found in the code comments for each class.

New Java files are create in the following way in Android Studio:


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
ContactBook app src main java swcguild com contactbook MainActivity
MainActivity.java
New
Cut Ctrl+X
Copy Ctrl+C
Copy Path Ctrl+Shift+C
Copy Reference Ctrl+Alt+Shift+C
Paste Ctrl+V
Find Usages Alt+F7
Find in Path... Ctrl+Shift+F
Replace in Path... Ctrl+Shift+R
Analyze
Refactor
Add to Favorites
Show Image Thumbnails Ctrl+Shift+T
Reformat Code... Ctrl+Alt+L
Optimize Imports... Ctrl+Alt+O
Delete... Delete
Make Module 'app' Ctrl+Shift+F9
Create Run Configuration
Run
Debug
Local History
Synchronize 'contactbook'
Show in Files
File Path Ctrl+Alt+Shift+2
Compare Directory with... Ctrl+D
Create Gist...
Java Class
Android resource file
Android resource directory
File
Package
Image Asset
AIDL
Activity
Folder
Fragment
Google
Other
Service
UI Component
Wear
Widget
XML
Terminal
6: Android
0: Messages
TODO
Event Log
Gradle Console
Memory Monitor
Create new Java class

```
package swcguild.com.contactbook;

import ...
```

New Java Class



Name: DatabaseConnector
Kind: Class

Create New Java Class Dialog for DatabaseConnector

DatabaseConnector Class

DatabaseConnector is a helper class used for interaction with the database. It uses a SQLiteDatabase object to manipulated the database and contains a nested class (**DatabaseOpenHelper**) to manage the creation and opening of the database.

DetailsFragment Class

This class is responsible for displaying one contact. It contains menu items that allow the user to either edit or delete the contact. It defines the DetailsFragment interface which contains callbacks for interacting with this class. These callback methods are implemented in the MainActivity class.

DetailsFragments defines an AsyncTask that it uses to retrieve a contact from the database in a thread separate from the UI thread.

AddEditFragment Class

This class is responsible for displaying the form for creating and editing contacts. It contains the logic for the Save button. This class defines the AddEditFragmentListener interface which contains callbacks for interacting with this class - these callback methods are implemented in the MainActivity class.

Like the DetailsFragment class above, this class defines an AsyncTask that is uses to write the contact information to the database using a thread separate from the UI thread.

ContactListFragment Class

This class is responsible for displaying a list of all the contacts in the database. It contains a menu item that allows the user to add a contact to the database. Like the previous two classes, it defines an interface (ContactListFragmentListerner) which contains callbacks for interacting with this class - these callbacks are implemented in MainActivity.

This class also defines an AsyncTask that is uses to retrieve all of the contacts from the database.

MainActivity Class

This class is responsible for managing the display of the various Fragment layouts. It implements all of the listener interfaces defined in our Fragment classes. As such, this class provides a link between various button/menu clicks and the functionality implemented in the Fragment classes.