Assignment A2 - Contact Manager

In this assignment, you'll create a simple contacts application. You'll work with the following concepts:

- GUI Components: Activities, TextViews, EditTexts, Layouts, RecyclerView, Fragments
- Data Storage: Databases and Content Providers

Prerequisites

This assignment assumes you have completed the following modules

- 1.2 Android Versions
- 1.3 Development Environment
- 1.5 Gradle Basics
- 1.7 Activities
- 2.1 Intents
- 2.2 RecyclerView and CardView
- 2.3 Toolbars
- 3.1 Fragments
- 3.2 Databases and Content Providers

THIS ASSIGNMENT WILL TAKE SOME TIME TO IMPLEMENT. READ THROUGH AND MAKE SURE YOU UNDERSTAND EVERYTHING REQUIRED AS SOON AS POSSIBLE. DO NOT WAIT UNTIL THE WEEK BEFORE THE ASSIGNMENT IS DUE TO START!!!

Overview

NOTE: Use the State-Based Fragment Framework approach discussed in class for this assignment

NOTE: The Content Provider version of the example is missing a line. Please add

```
setHasStableIds(true);
```

to your adapter's constructor. This will cause the base adapter class in RecyclerView to grab the id for the row and set it in the ViewHolder.

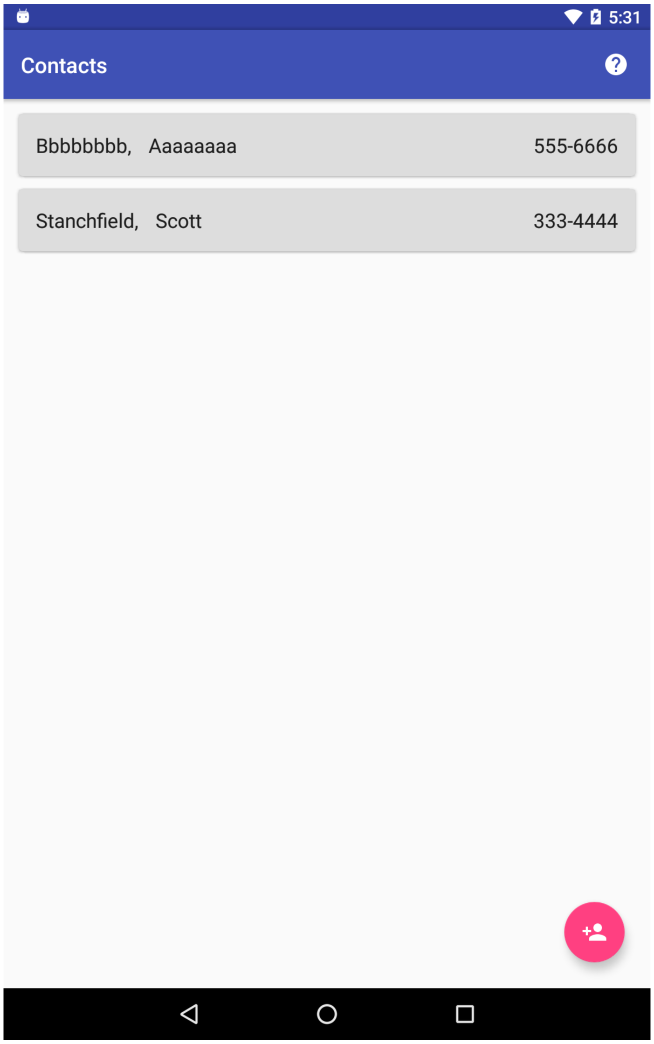
For this assignment, you'll need to create:

- A Contact class
    - id (long)
    - First and Last Name (separate Strings)
    - Phone numbers (Strings): Home, Work, Mobile
    - Email Address (String)
- ContactsActivity
    - Manages fragments
    - Has an action bar with an "about" action (launches AboutActivity via an explicit intent)
- AboutActivity - shows simple "about" text
- Three Fragments
    - Contacts Fragment
        - RecyclerView of all contacts
            - Display last name, first name, and mobile phone in each row
        - Use CursorLoader to load the data
        - When an item is tapped, display it in the Display fragment
        - When an item is swiped, delete it
        - Use a Floating Action Button to add new contacts to the list (opens EditFragment)
    - DisplayFragment
        - Scrolling layout of views to display a contact's data
        - Action bar
            - Email: uses an implicit intent to send an email
            - Edit: shows the EditFragment with the current contact
    - EditFragment
        - Scrolling layout of views to edit a contact's data
        - Action bar
            - Done: saves the contact and returns

    - Cancel: does not save, and returns
    - Reload: reloads the original contact data
- Content provider to manage the contact data

Sample Screenshots

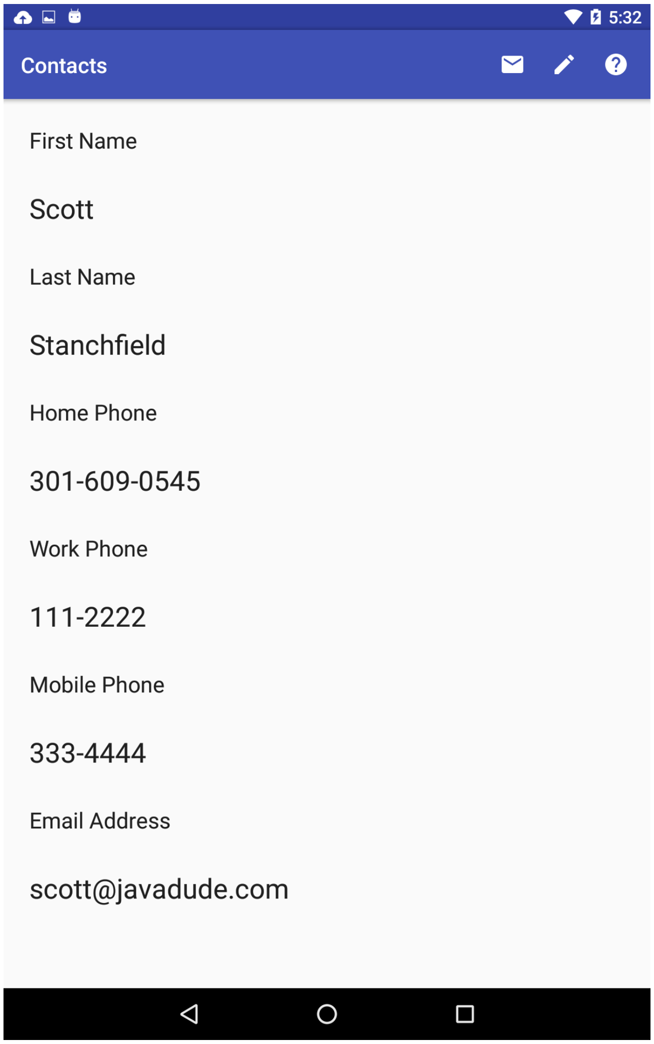
Feel free to change the layouts of styles as you see fit.



Bbbbbbbb, Aaaaaaaa 555-6666

Stanchfield, Scott 333-4444

Figure 1 - Contact List (portrait)



First Name

Scott

Last Name

Stanchfield

Home Phone

301-609-0545

Work Phone

111-2222

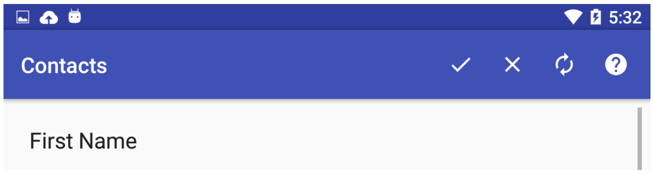
Mobile Phone

333-4444

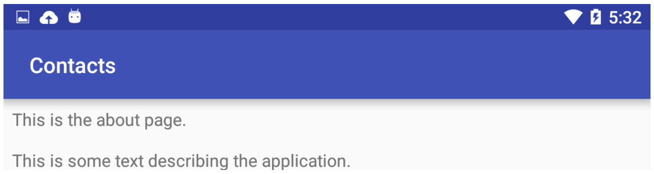
Email Address

scott@javadude.com

Figure 2 - Display Contact (portrait)

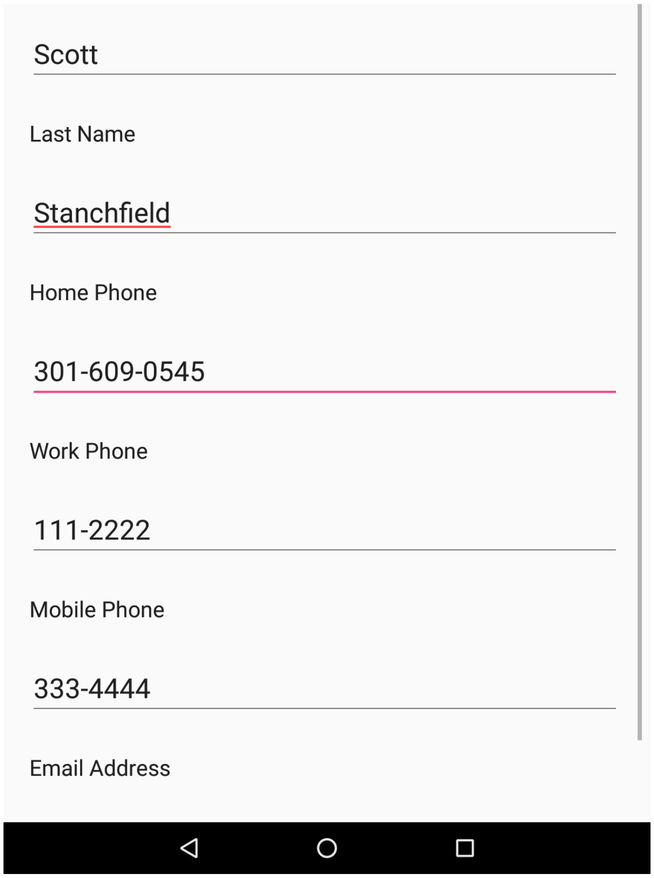


First Name



This is the about page.

This is some text describing the application.



Scott

Last Name

Stanchfield

Home Phone

301-609-0545

Work Phone

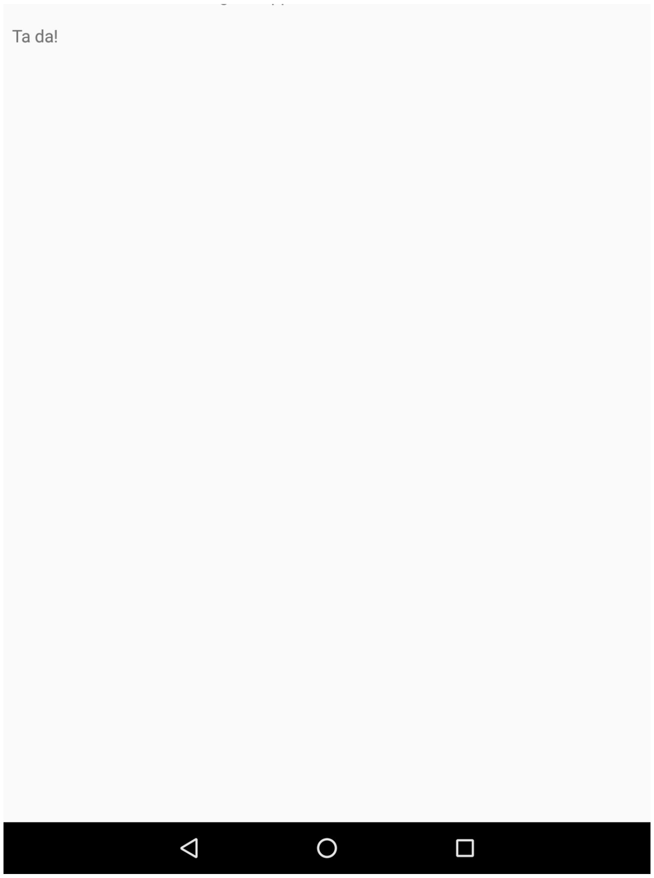
111-2222

Mobile Phone

333-4444

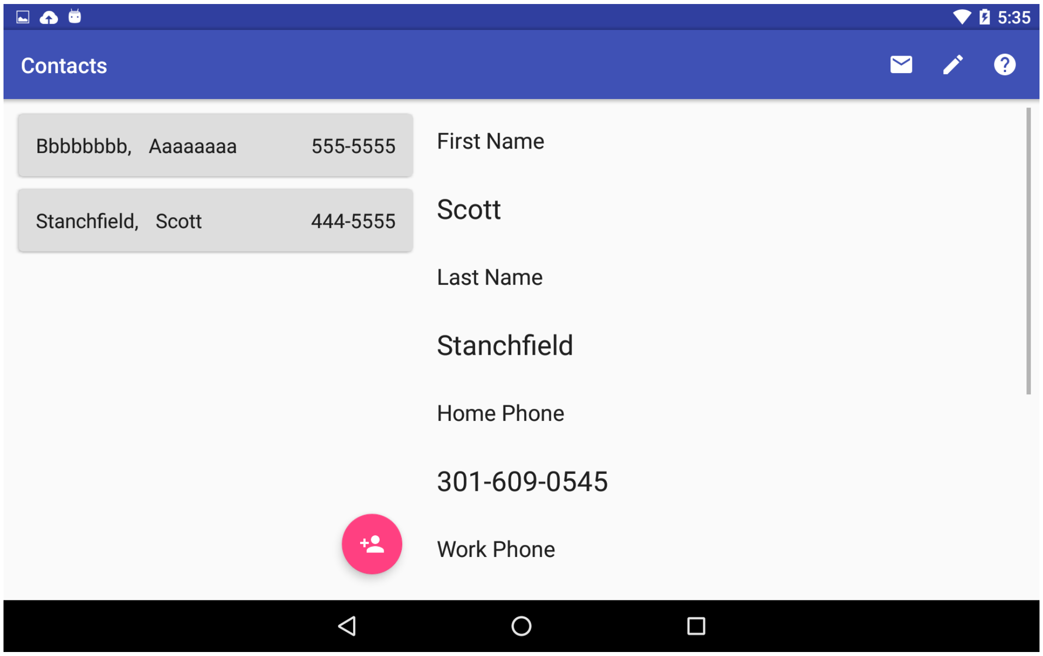
Email Address

Figure 3 - Edit Contact (portrait)



Ta da!

Figure 4 - About Activity (portrait)



Bbbbbbbb, Aaaaaaaa	555-5555
Stanchfield, Scott	444-5555

First Name

Scott

Last Name

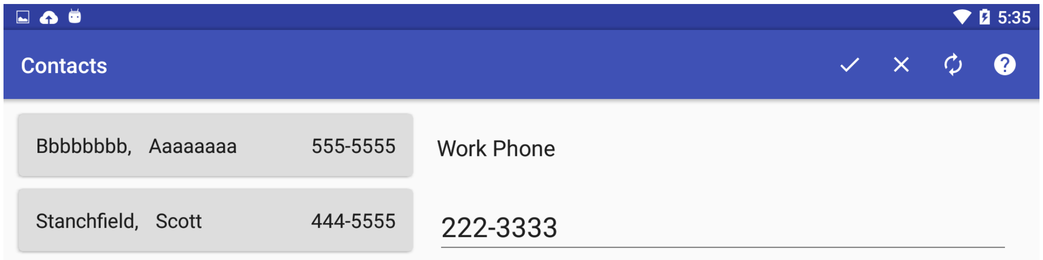
Stanchfield

Home Phone

301-609-0545

Work Phone

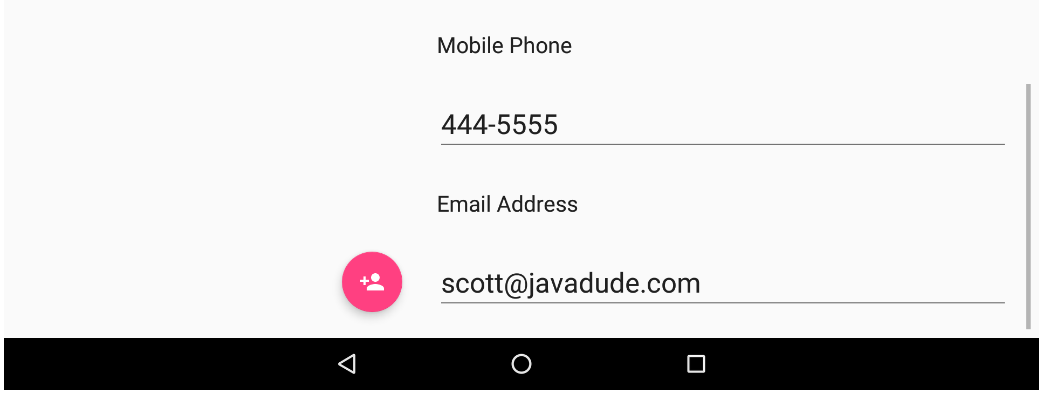
Figure 5 - List and Display (landscape)



Bbbbbbbb, Aaaaaaaa	555-5555
Stanchfield, Scott	444-5555

Work Phone

222-3333



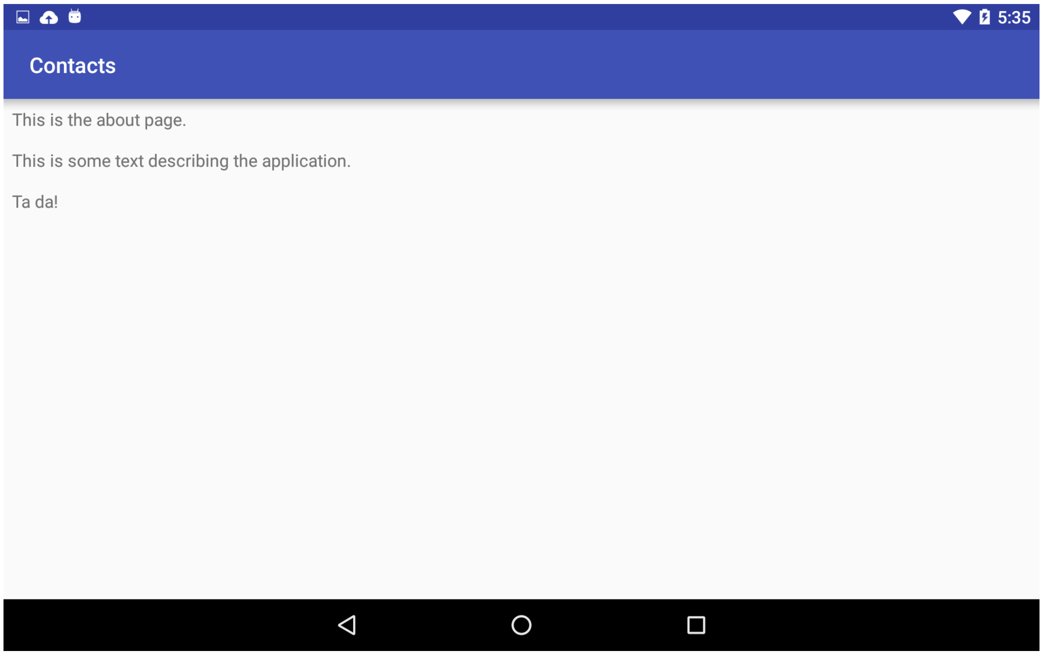
Mobile Phone

444-5555

Email Address

scott@javadude.com

Figure 6 - List and Edit (landscape)



Contacts

This is the about page.

This is some text describing the application.

Ta da!

Figure 7 - About Activity (landscape)

General Notes

Use the State-Based Fragment Framework approach discussed in class for this assignment. You should only define two activities for your submission (the ContactsActivity and the AboutActivity). (Note: The FragmentFrameworkActivity should be included with the submission as well)

Be sure to name this project HW2 and its packages starting with lastname.firstname.hw2.

For your EditTexts and TextViews, please use the following ids so we can set up some automated testing:

- R.id.first_name
- R.id.last_name
- R.id.home_phone

- R.id.work_phone
- R.id.mobile_phone
- R.id.email
- NOTE: in your RecyclerView's list item view, use separate TextViews for each part:
    - first_name
    - last_name
    - mobile_phone
- use R.id.fragment_container1 and R.id.fragment_container2 for the ids of your fragment containers

For your database name use CONTACTS with table CONTACTS containing the following columns:

- _id
- FIRST_NAME
- LAST_NAME

- LAST_NAME
- HOME_PHONE
- WORK_PHONE
- MOBILE_PHONE
- EMAIL

Generate icons in Android Studio as demonstrated in class. Use the icons shown in the screenshots above. I recommend you use vector icons and change the colors to white

Start with the state-based Todo example demonstrated in class. This is similar to your application, but you will have the additional DisplayFragment.

Choose appropriate inputTypes for the fields (phone numbers, email, etc)

Note: Use an ***explicit*** intent to open the AboutActivity; use an ***implicit*** intent to send email.

Hardcoded data (such as field labels) ***must not*** appear in the xml layout or Java classes unless I ok it in class; you must use the strings.xml file. (This note will not be repeated in later assignments but still applies).

To start another application to handle email:

```
Intent emailIntent = new Intent(Intent.ACTION_SENDTO,
        Uri.fromParts("mailto", email, null));
emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Hi " + name + "!");
emailIntent.putExtra(Intent.EXTRA_TEXT, "Just wanted to say hi...");
startActivity(emailIntent);
```

Content Provider Notes

You do not need to set up the code to perform database upgrades! This assignment acts as version 1 of your database. (If you're following the example code, be sure to remove the extra work done during database update...)

Create a content provider to manager your contacts, similar to the 'todo' content provider demonstrated in class.

When creating your database, be sure to name your id field _id as described in class (integer primary key autoincrement)

Be sure the only way that contacts are stored/retrieved is through the contact provider (using getContentManager() or the CursorLoader)

Submitting Your Assignment

To submit your assignment (do this outside Android Studio):

1. Close the project in Android Studio (and/or quit Android Studio)
2. Find the directory for your project
3. Delete the build and app/build directories from the project folder
4. Zip the project directory. Be sure to get all files and subdirectories, including those that start with '.'. Be sure to name the zip file lastname.firstname.hw2
5. Go to the Blackboard course website
6. Click on the ***Assignment Submission*** link on the left navigation bar
7. Click on the ***Assignment A2*** link
8. Attach your submission and enter any notes
9. Be sure to ***submit*** your assignment, not just save it!

Content License

Source Code License