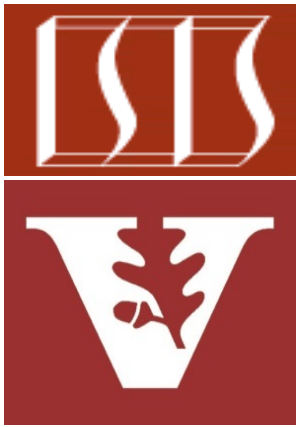
Introduction to Android Activities (Part 1)

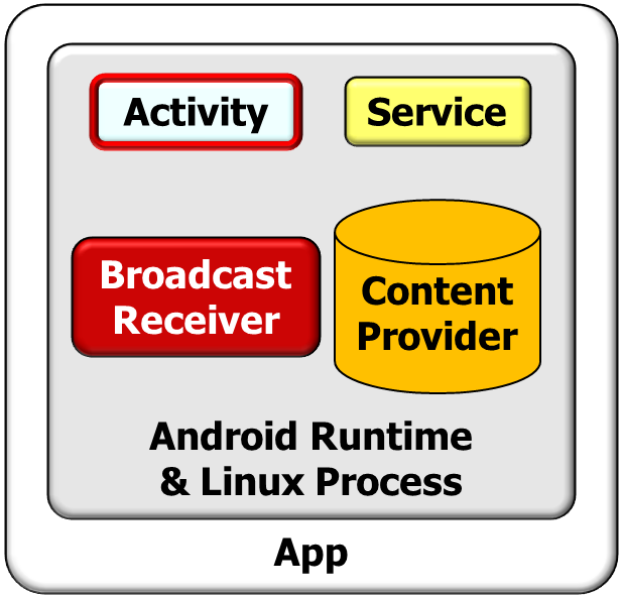


Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA

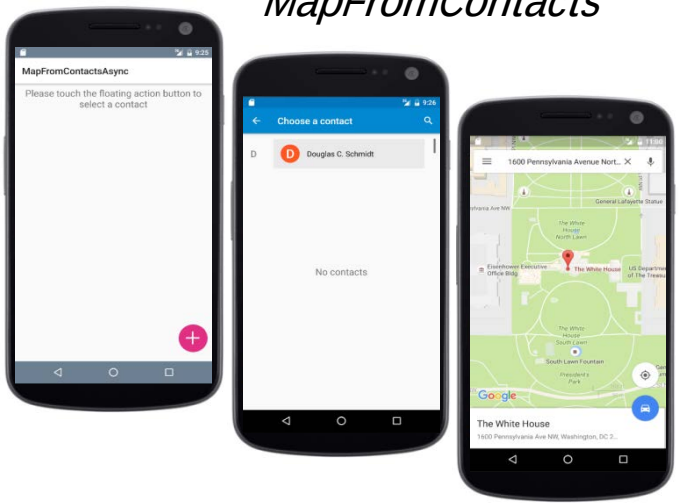
Learning Objectives in this Part of the Lesson

- Know how an activity defines a user-facing operation that's displayed on a device screen

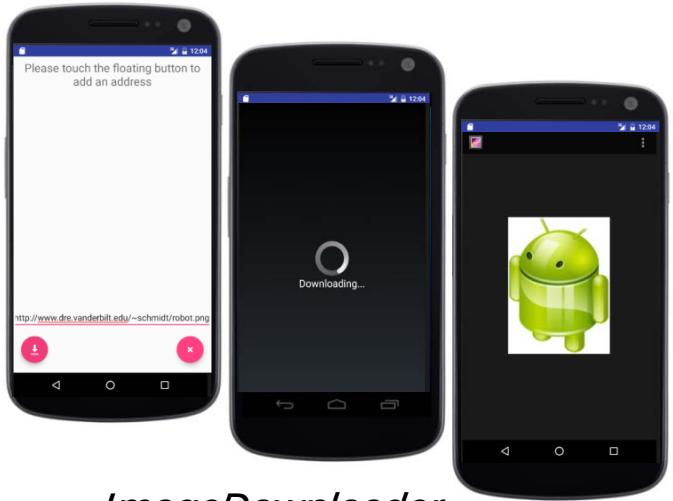


Learning Objectives in this Part of the Lesson

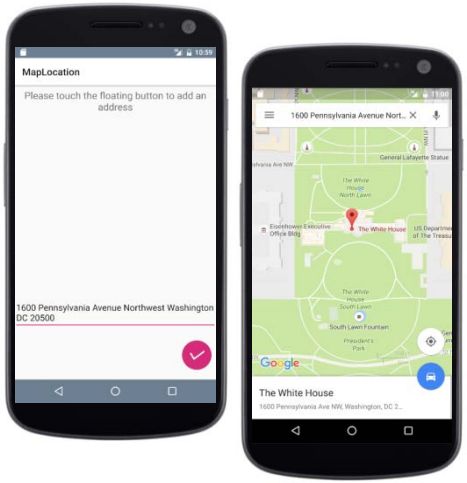
- Know how an activity defines a user-facing operation that's displayed on a device screen
- Recognize the various apps used as case studies throughout this module



MapFromContacts



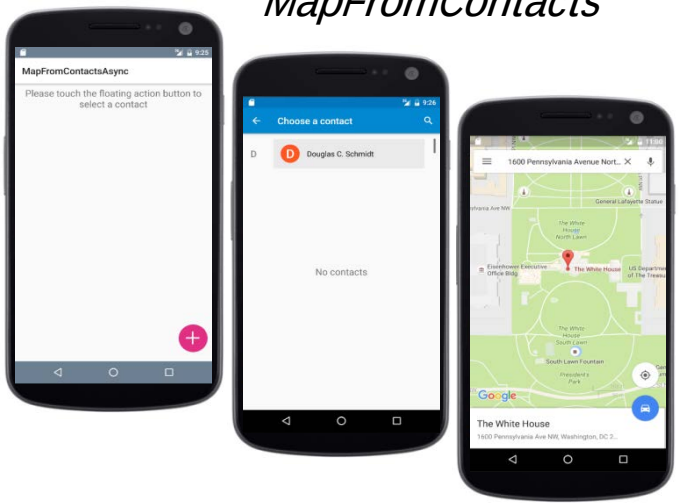
ImageDownloader



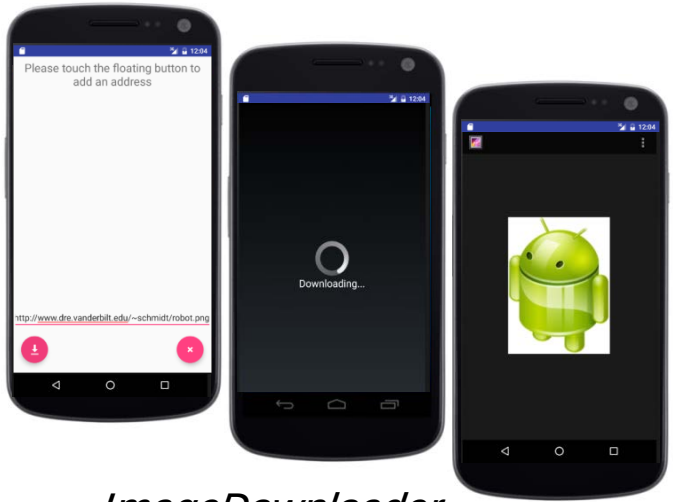
MapLocation

Learning Objectives in this Part of the Lesson

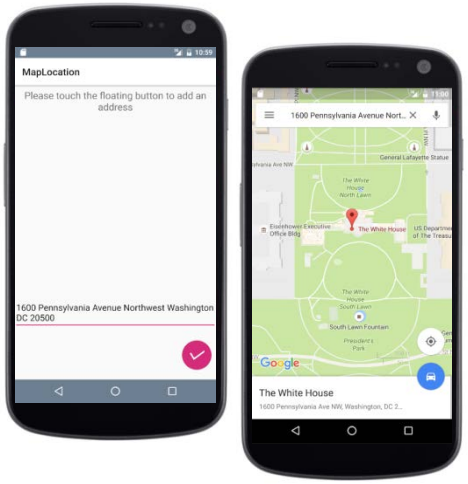
- Know how an activity defines a user-facing operation that's displayed on a device screen
- Recognize the various apps used as case studies throughout this module



MapFromContacts



ImageDownloader



MapLocation

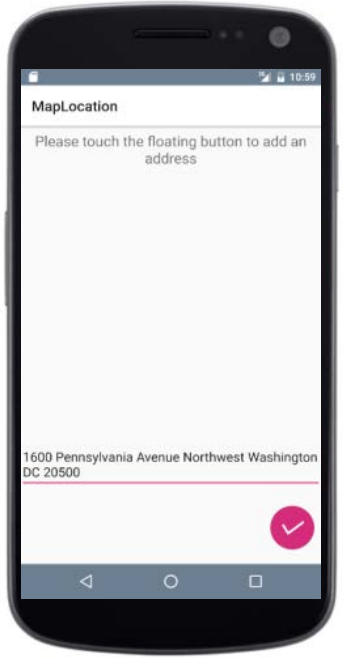
The source code for all these apps is available at www.gitlab.com

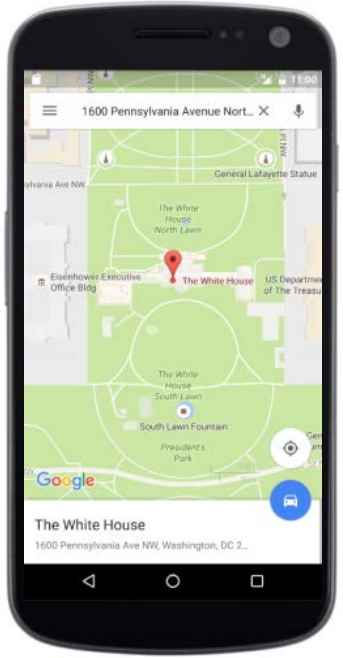
Learning Objectives in this Part of the Lesson

- Know how an activity defines a user-facing operation that's displayed on a device screen
- Recognize the various apps used as case studies throughout this module
- Understand the general steps used to implement an activity

Learning Objectives in this Part of the Lesson

- Know how an activity defines a user-facing operation that's displayed on a device screen
- Recognize the various apps used as case studies throughout this module
- Understand the general steps used to implement an activity



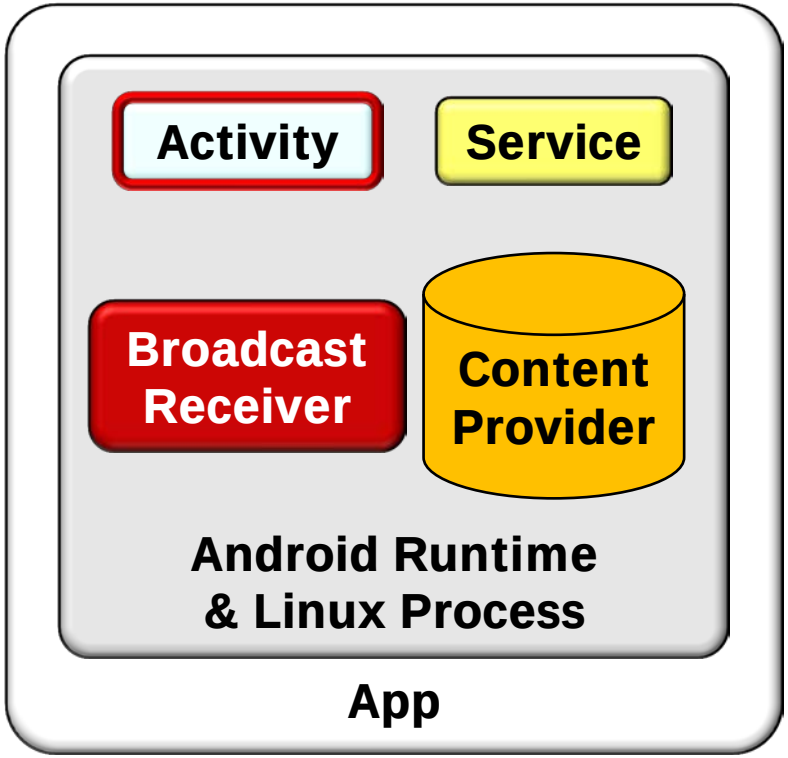


Part 2 will then show how these steps are applied in the MapLocation app

Overview of Android Activities

Overview of Android Activities

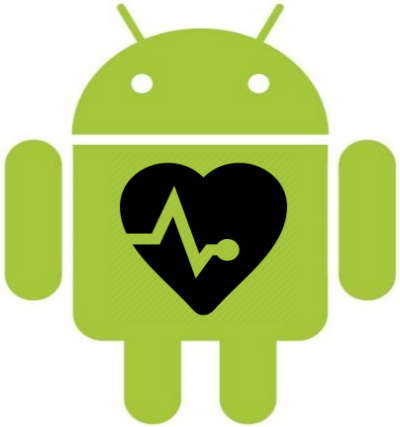
- An Activity is the most common type of Android component

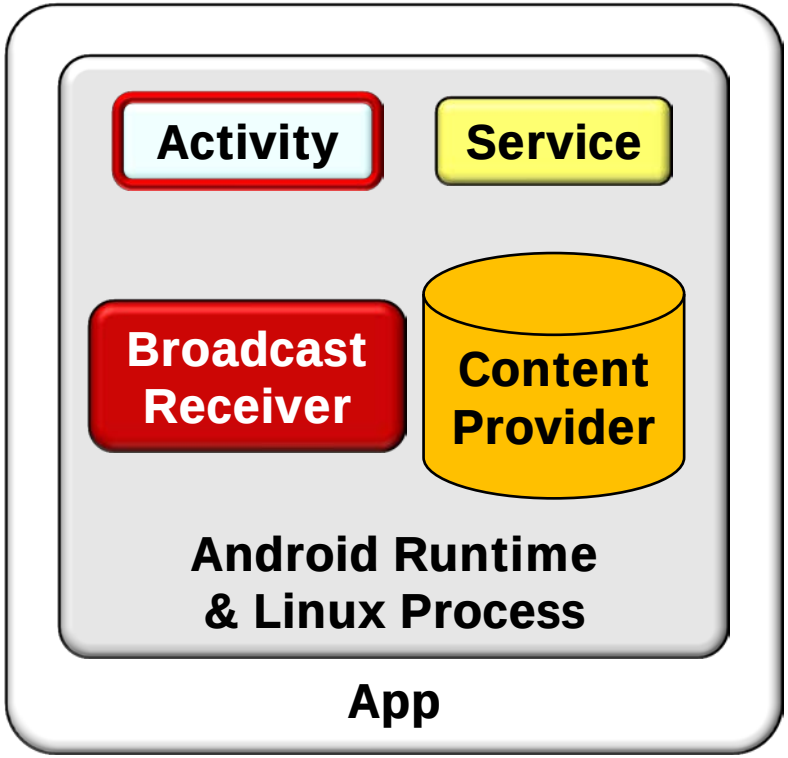


See developer.android.com/reference/android/app/Activity.html

Overview of Android Activities

- An Activity is the most common type of Android component
  - It's at the heart of all Android apps

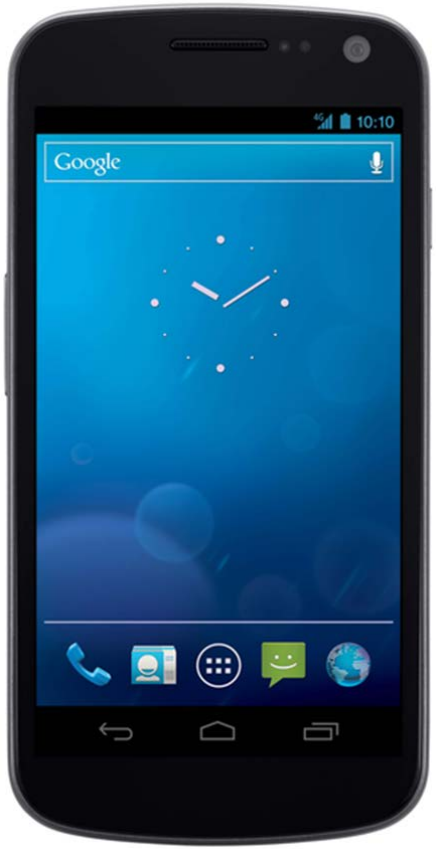




See developer.android.com/reference/android/app/Activity.html

Overview of Android Activities

- An activity defines a user-facing operation that's displayed on a device screen

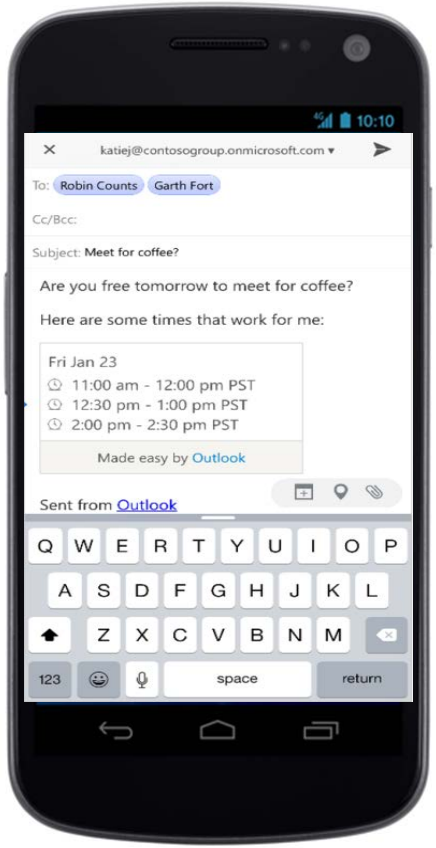


Overview of Android Activities

- An activity defines a user-facing operation that's displayed on a device screen, e.g.
  - Showing a login prompt

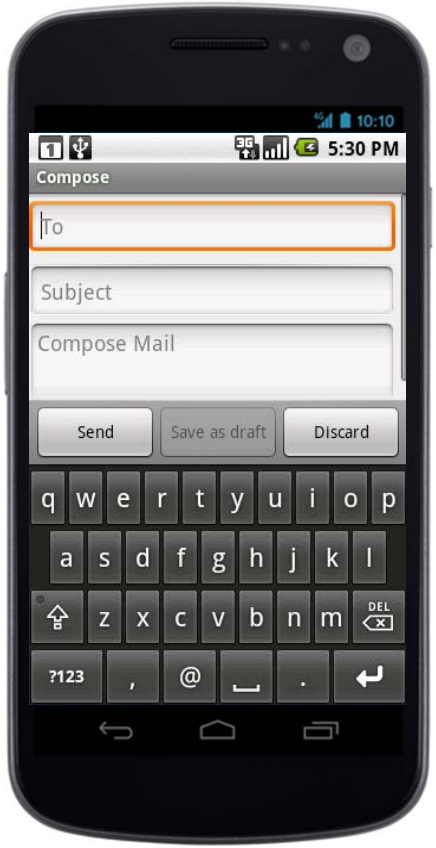
Overview of Android Activities

- An activity defines a user-facing operation that's displayed on a device screen, e.g.
  - Showing a login prompt
  - Reading an email message



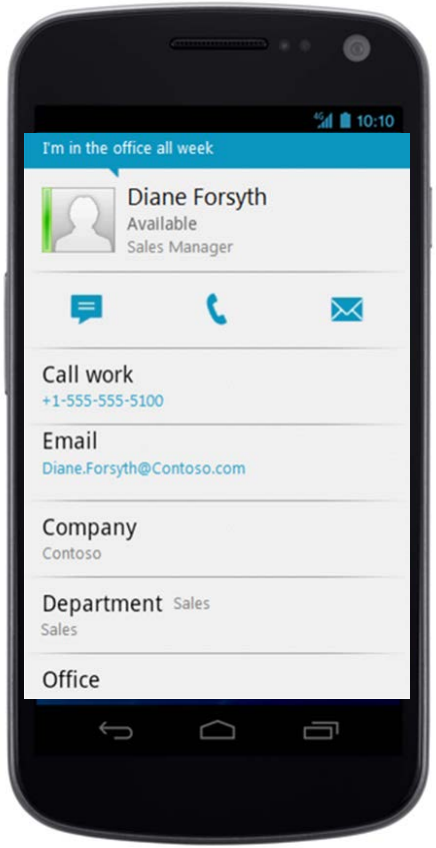
Overview of Android Activities

- An activity defines a user-facing operation that's displayed on a device screen, e.g.
  - Showing a login prompt
  - Reading an email message
  - Composing an email message



Overview of Android Activities

- An activity defines a user-facing operation that's displayed on a device screen, e.g.
  - Showing a login prompt
  - Reading an email message
  - Composing an email message
  - Viewing a contact

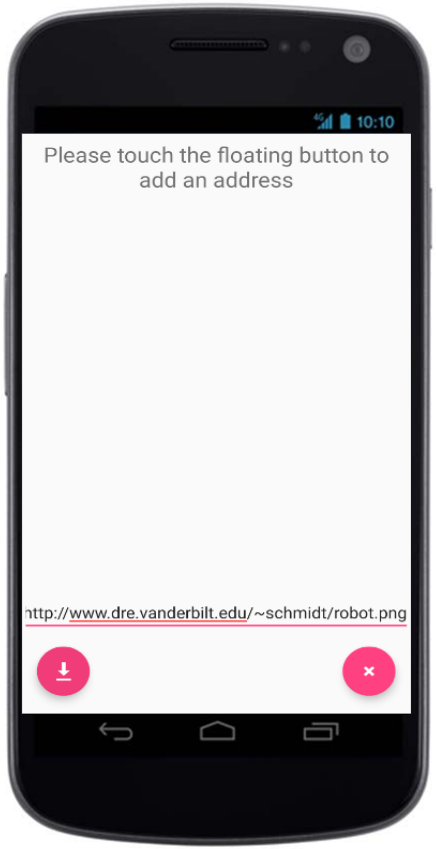


Overview of Android Activities

- An activity defines a user-facing operation that's displayed on a device screen, e.g.
  - Showing a login prompt
  - Reading an email message
  - Composing an email message
  - Viewing a contact
  - Browsing the Internet

Overview of Android Activities

- An activity defines a user-facing operation that's displayed on a device screen, e.g.
  - Showing a login prompt
  - Reading an email message
  - Composing an email message
  - Viewing a contact
  - Browsing the Internet
  - Downloading



Overview of Android Activities

- An activity defines a user-facing operation that's displayed on a device screen, e.g.
  - Showing a login prompt
  - Reading an email message
  - Composing an email message
  - Viewing a contact
  - Browsing the Internet
  - Downloading & displaying an image

Overview of Android Activities

- An activity defines a user-facing operation that's displayed on a device screen, e.g.
  - Showing a login prompt
  - Reading an email message
  - Composing an email message
  - Viewing a contact
  - Browsing the Internet
  - Downloading & displaying an image

Activities needn't provide a user-facing operation, though this is a common use-case

Overview of Android Activities

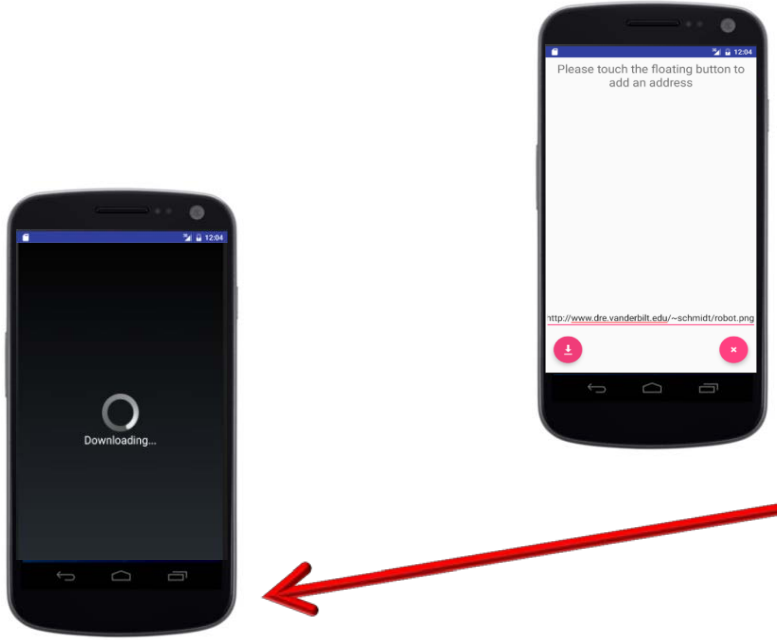
- An Android app may contains more than one activity

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="DownloadImageViewer"
    android:supportsRtl="true"
    android:theme="@style/AppTheme" >
    <activity
        android:name=".activities.MainActivity"
        android:label="DownloadImageViewer"
        android:windowSoftInputMode="adjustPan" >
        <intent-filter...>
    </activity>

    <activity
        android:name=".activities.DownloadImageActivity"
        android:theme="@android:style/Theme.NoTitleBar">
        <intent-filter...>
    </activity>

    <provider...>
</application>
```

See developer.android.com/guide/components/activities.html

Overview of Android Activities

- An Android app may contains more than one activity
  - These activities are described in the AndroidManifest.xml file

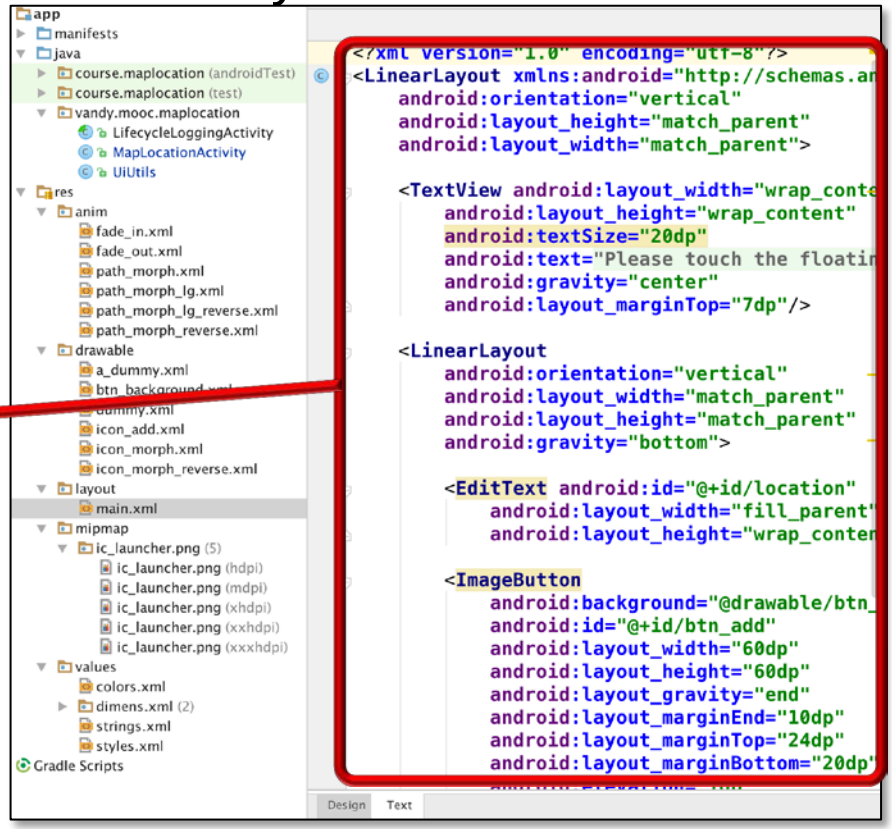



```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="DownloadImageViewer"
    android:supportsRtl="true"
    android:theme="@style/AppTheme" >
    <activity
        android:name=".activities.MainActivity"
        android:label="DownloadImageViewer"
        android:windowSoftInputMode="adjustPan" >
        <intent-filter...>
    </activity>

    <activity
        android:name=".activities.DownloadImageActivity"
        android:theme="@android:style/Theme.NoTitleBar">
        <intent-filter...>
    </activity>

    <provider...>
</application>
```

See developer.android.com/guide/topics/manifest/manifest-intro.html

Overview of Android Activities

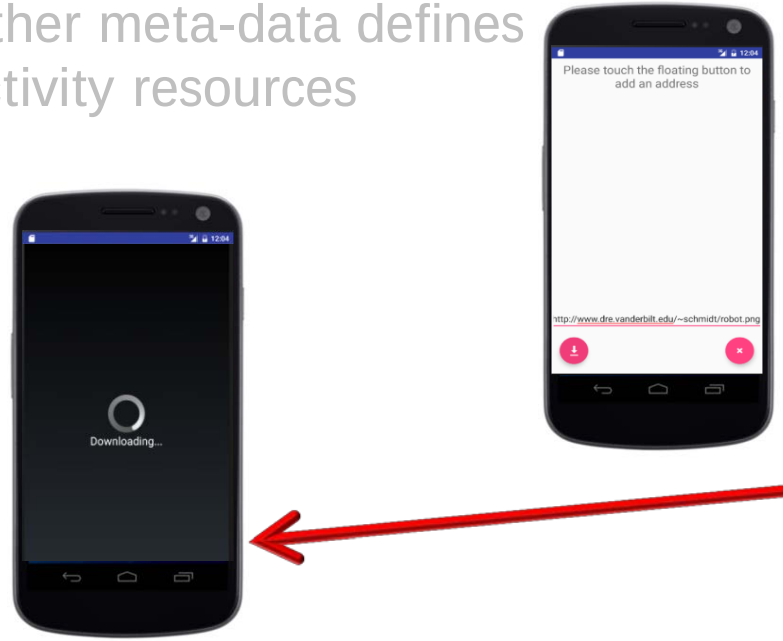
- An Android app may contains more than one activity
  - These activities are described in the AndroidManifest.xml file
  - Other meta-data defines activity resources



See developer.android.com/guide/topics/resources/accessing-resources.html

Overview of Android Activities

- An Android app may contains more than one activity
  - These activities are described in the AndroidManifest.xml file
  - Other meta-data defines activity resources



DownloadImageActivity.java

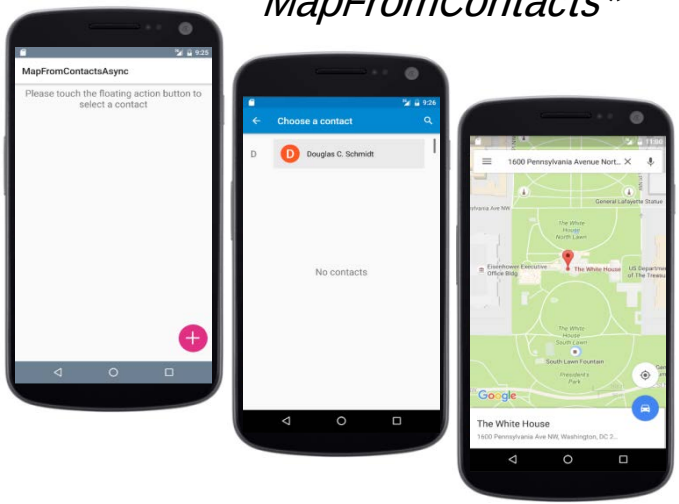
```
    */
   @Override
   protected void onStart() {
       //...
       super.onStart();

       // Make progress bar visible.
       mLoadingProgressBar.setVisibility(View.VISIBLE);

       Thread thread = mRetainedFragmentManager.get(THREAD);

       if (thread == null) {
           Log.d(TAG,
                   "onStart() creating and executing a Thread");

           //...
           Runnable DownloadImageTask = () -> {
               //...
               final Uri imagePath =
               DownloadUtils.downloadImage(DownloadImageActivity.this,
                       mRetainedFragmentManager.get(URL));

               // Set the result of the Activity.
               UiUtils.setActivityResult(DownloadImageActivity.this,
                                         imagePath,
                                         "download failed");

               // Run finish() on the UI Thread.
               DownloadImageActivity.this.runOnUiThread(() -> {
                       // This lambda runs on the UI thread.
                       Log.d(TAG, "runOnUiThread()");

                       // Stop the Activity from running.
                       DownloadImageActivity.this.finish();
                   });
           };
```

Activity code itself is typically written in Java

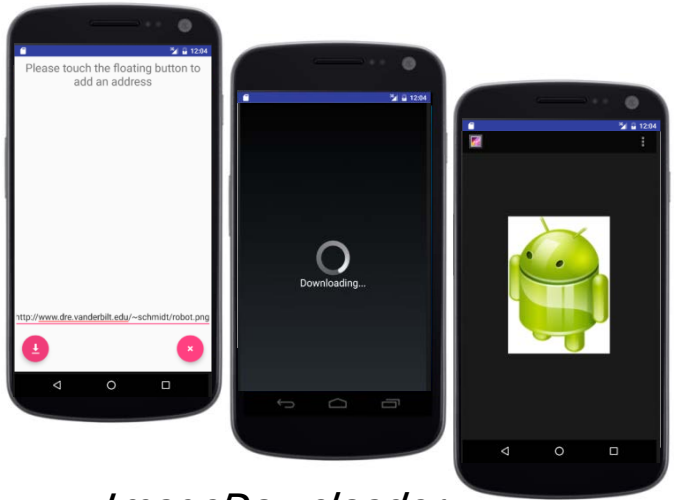
Overview of Case Study Apps

Overview of Case Study Apps

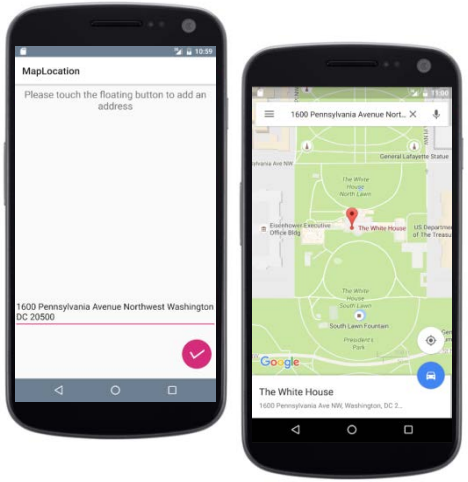
- Throughout this module we'll illustrate key elements & properties of activities via several representative apps



*MapFromContacts**



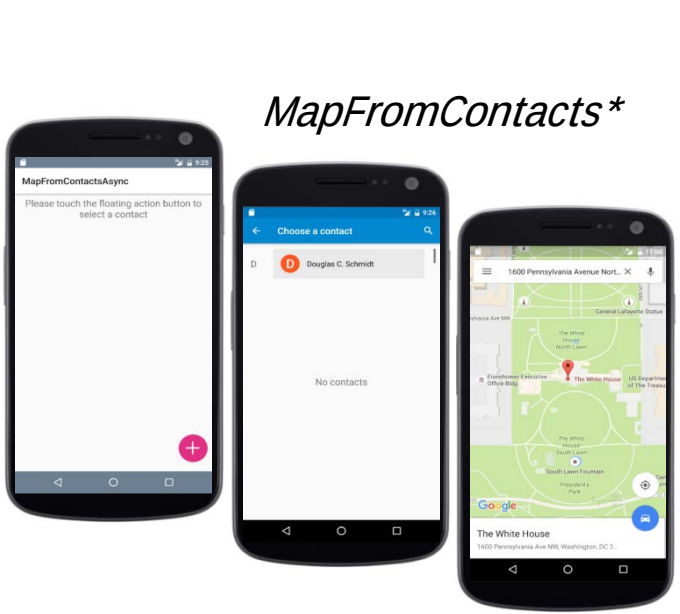
ImageDownloader



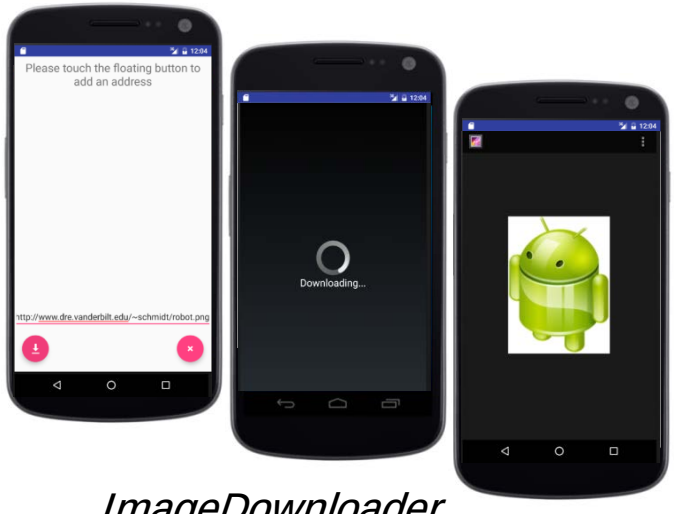
MapLocation

Overview of Case Study Apps

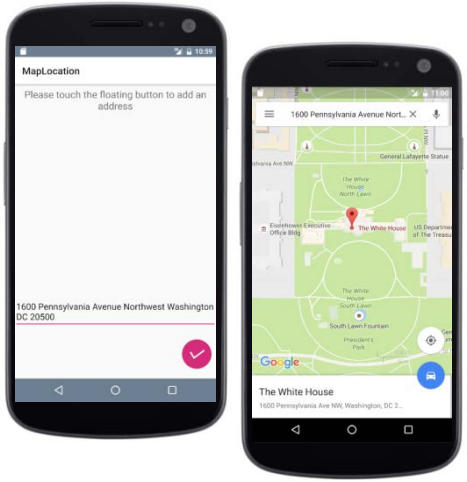
- Throughout this module we'll illustrate key elements & properties of activities via several representative apps
  - One app will look familiar, but the others are new



*MapFromContacts**



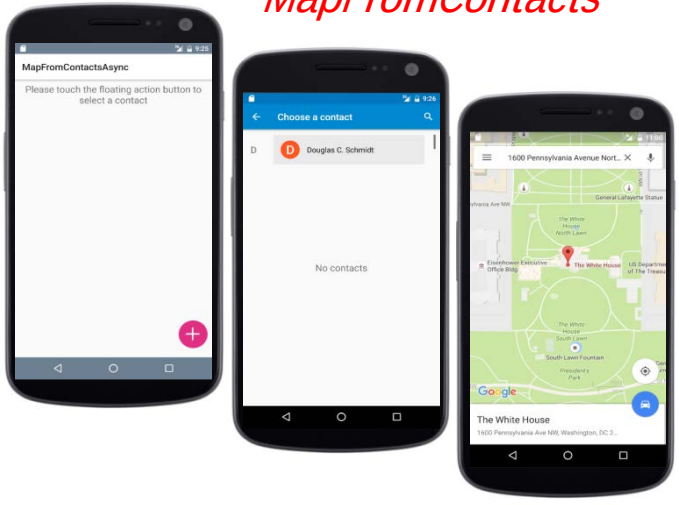
ImageDownloader



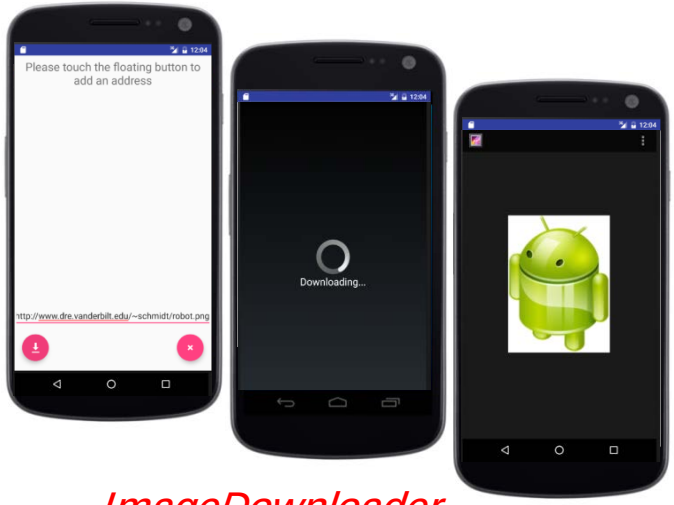
MapLocation

Overview of Case Study Apps

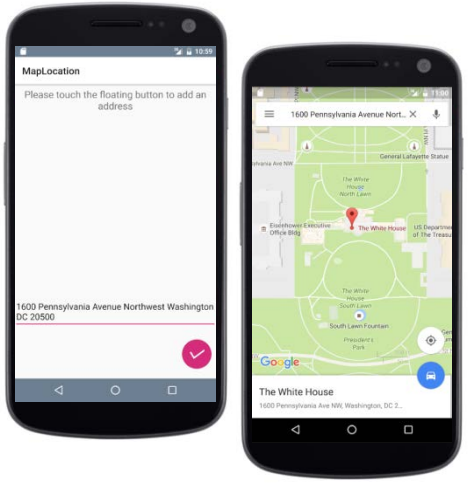
- Throughout this module we'll illustrate key elements & properties of activities via several representative apps
  - One app will look familiar, but the others are new



*MapFromContacts**



ImageDownloader

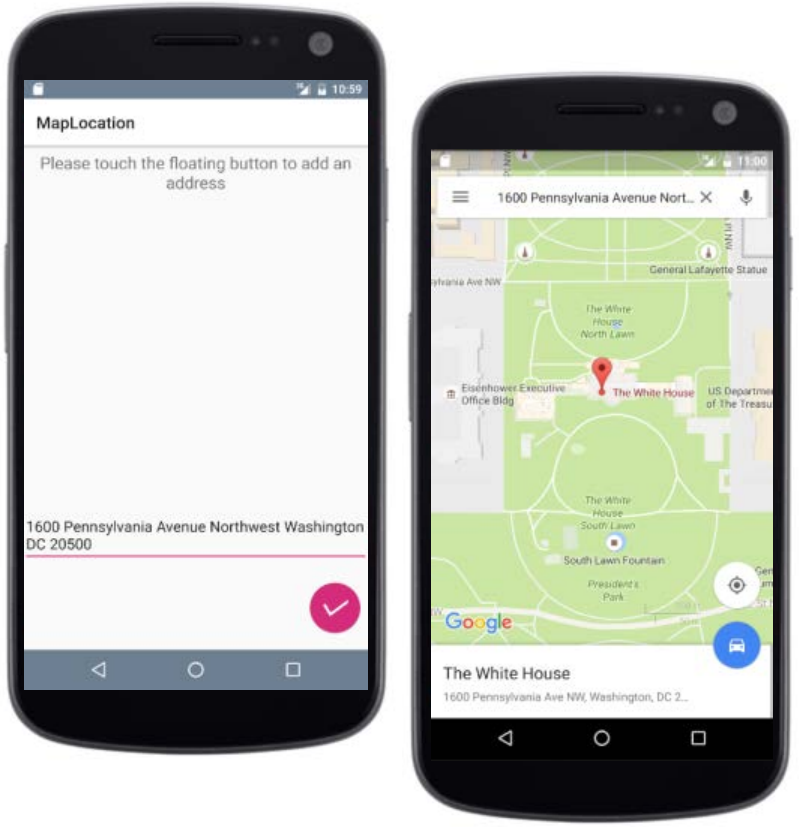


MapLocation

Source code for all these app case studies is available at gitlab.com/vandy-aad-2

Overview of MapLocation

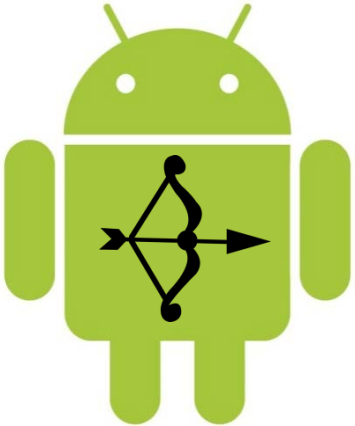
- This app uses an intent & an activity to map a location from an address given by the user

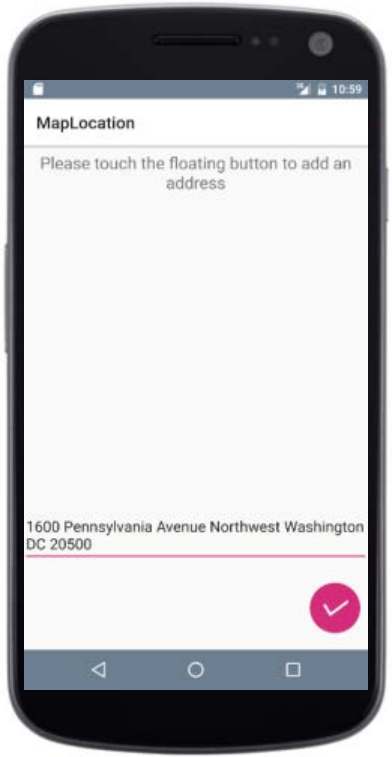


See gitlab.com/vandy-aad-2/MapLocation

Overview of MapLocation

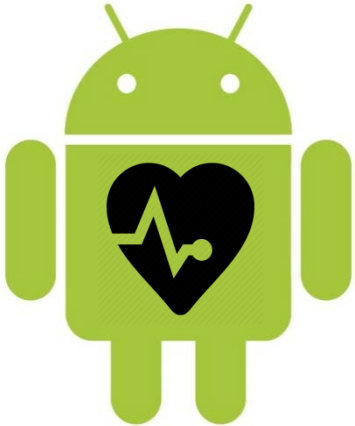
- This app uses an intent & an activity to map a location from an address given by the user
  - Earlier we reviewed this app's code wrt its use of intents

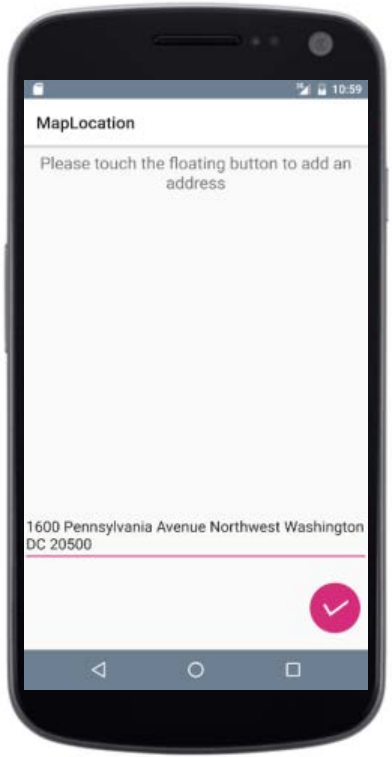




Overview of MapLocation

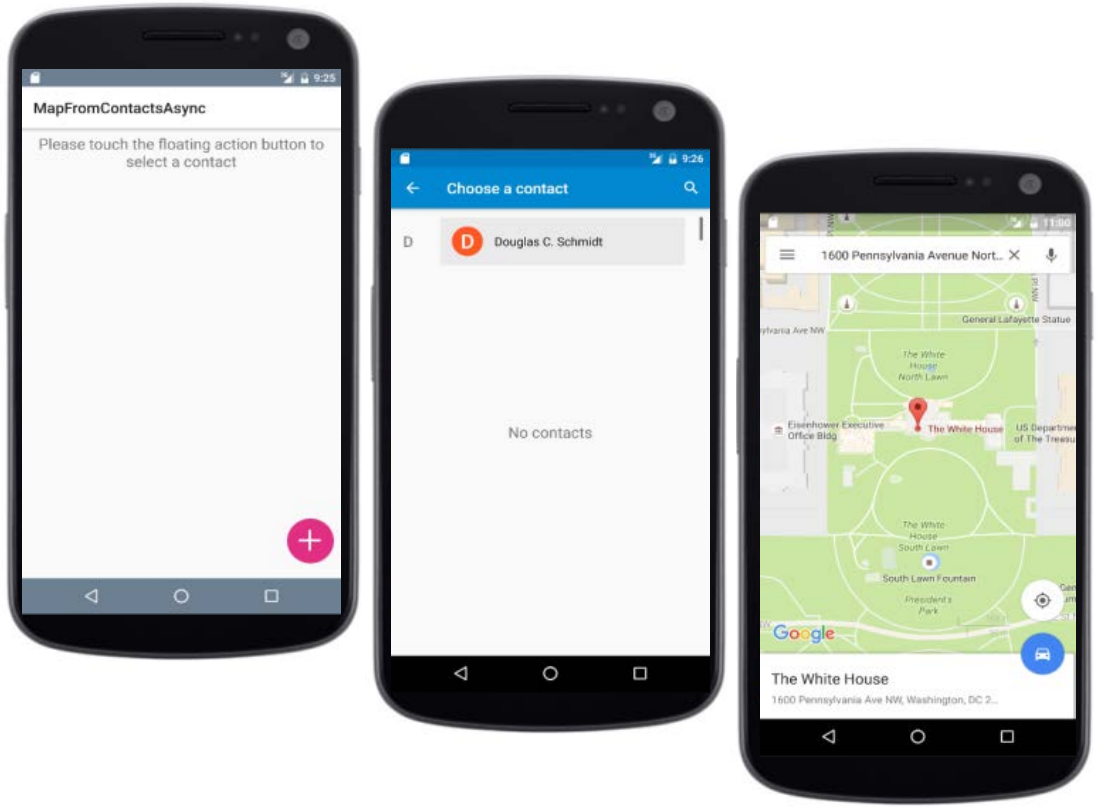
- This app uses an intent & an activity to map a location from an address given by the user
  - Earlier we reviewed this app's code wrt its use of intents
  - Now we'll review it wrt its use of activities





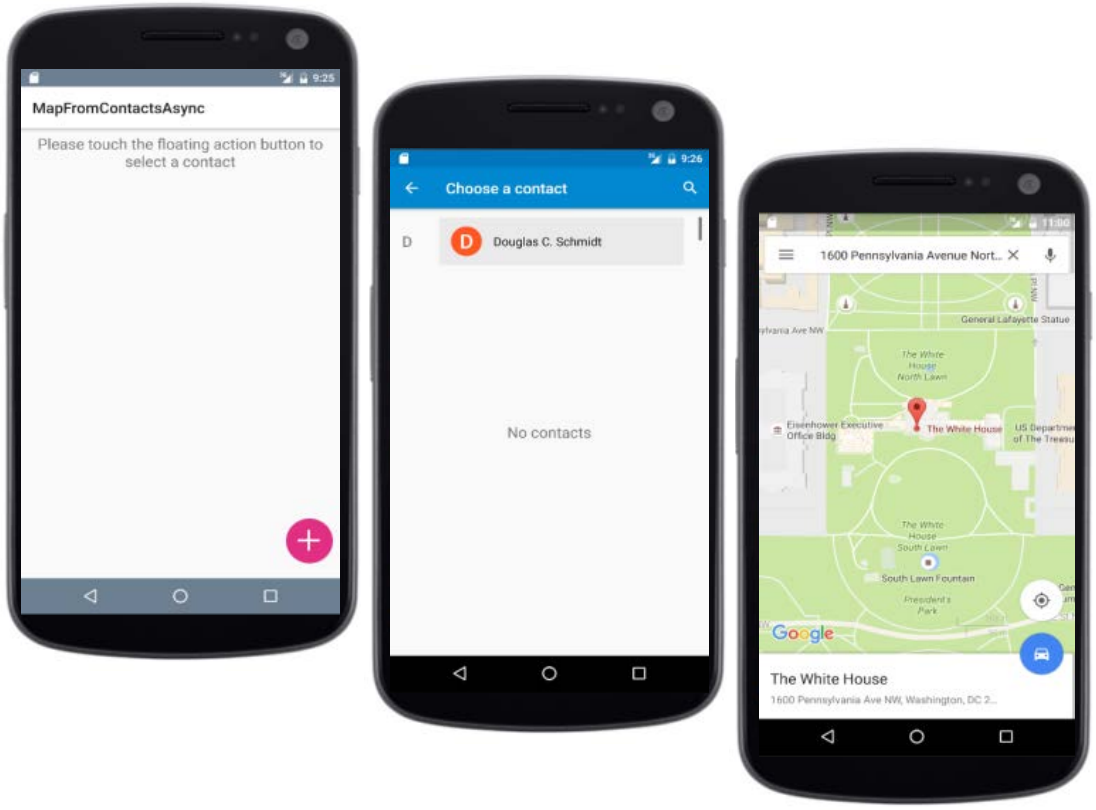
Overview of MapFromContacts*

- These apps uses an intent & an activity to map the address of a contact using Android concurrency frameworks



Overview of MapFromContacts*

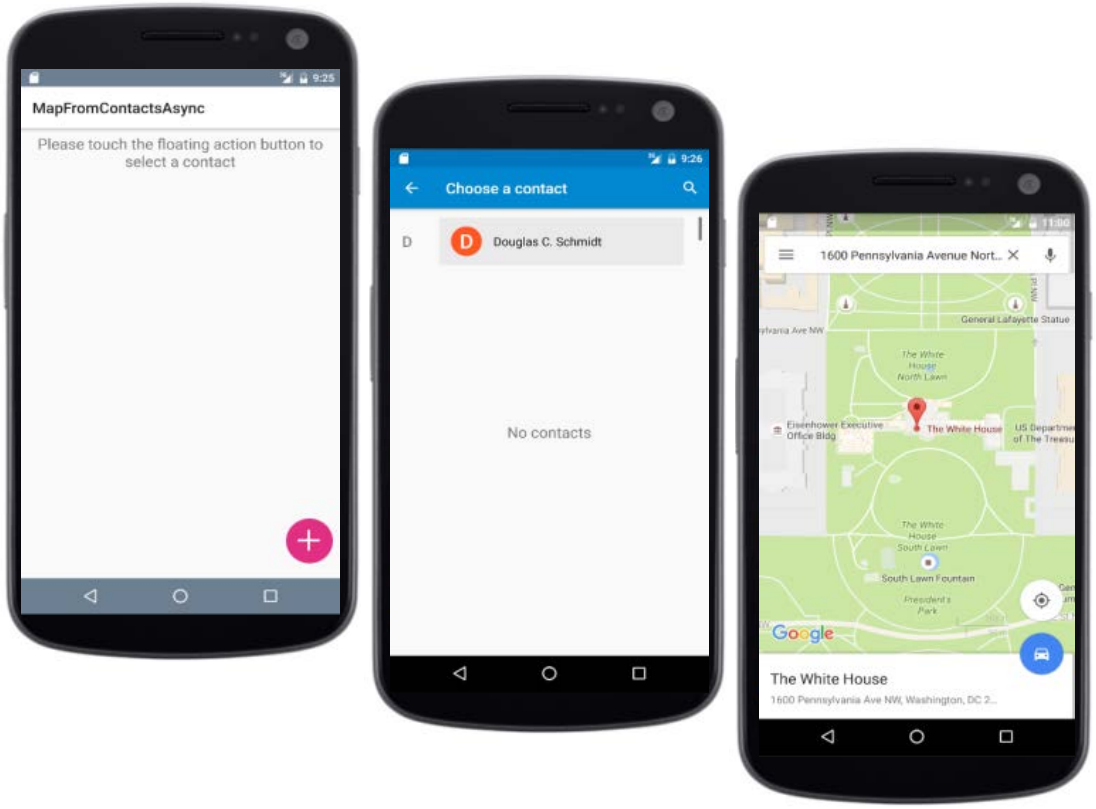
- These apps uses an intent & an activity to map the address of a contact using Android concurrency frameworks, e.g.
  - HaMeR framework



See gitlab.com/vandy-aad-2/MapFromContactsHaMeR

Overview of MapFromContacts*

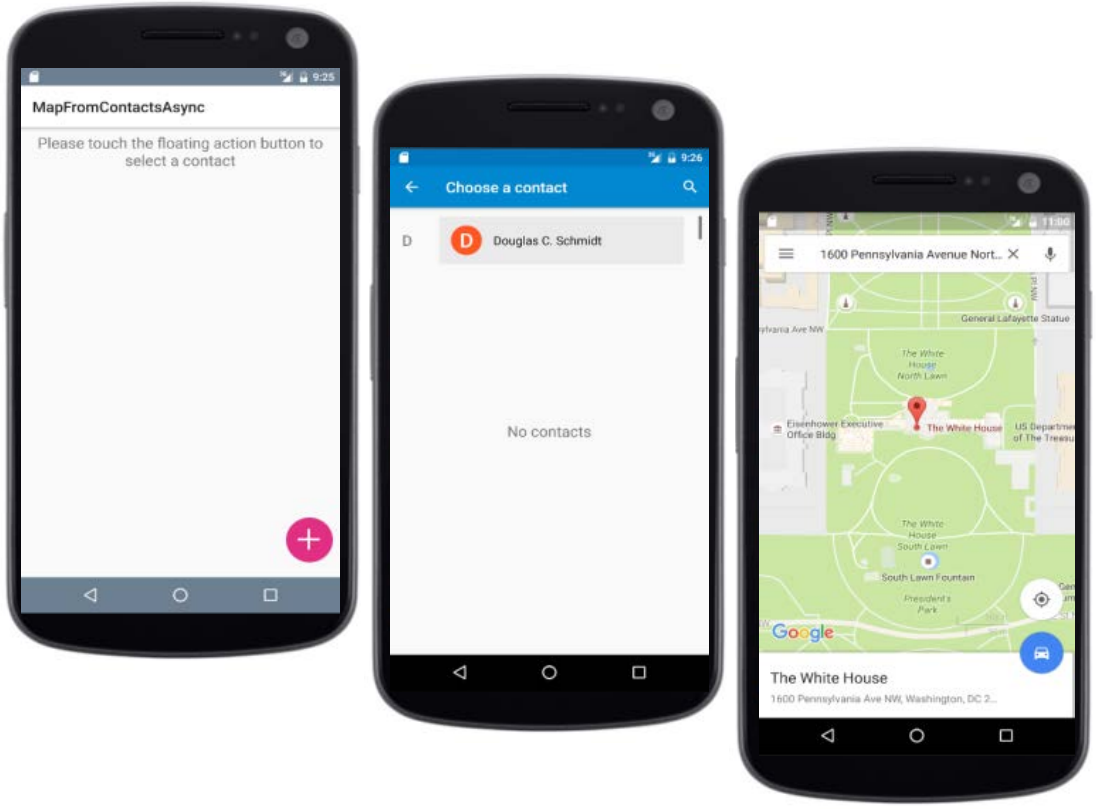
- These apps uses an intent & an activity to map the address of a contact using Android concurrency frameworks, e.g.
  - HaMeR framework
  - AsyncTask framework



See gitlab.com/vandy-aad-2/MapFromContactsAsync

Overview of MapFromContacts*

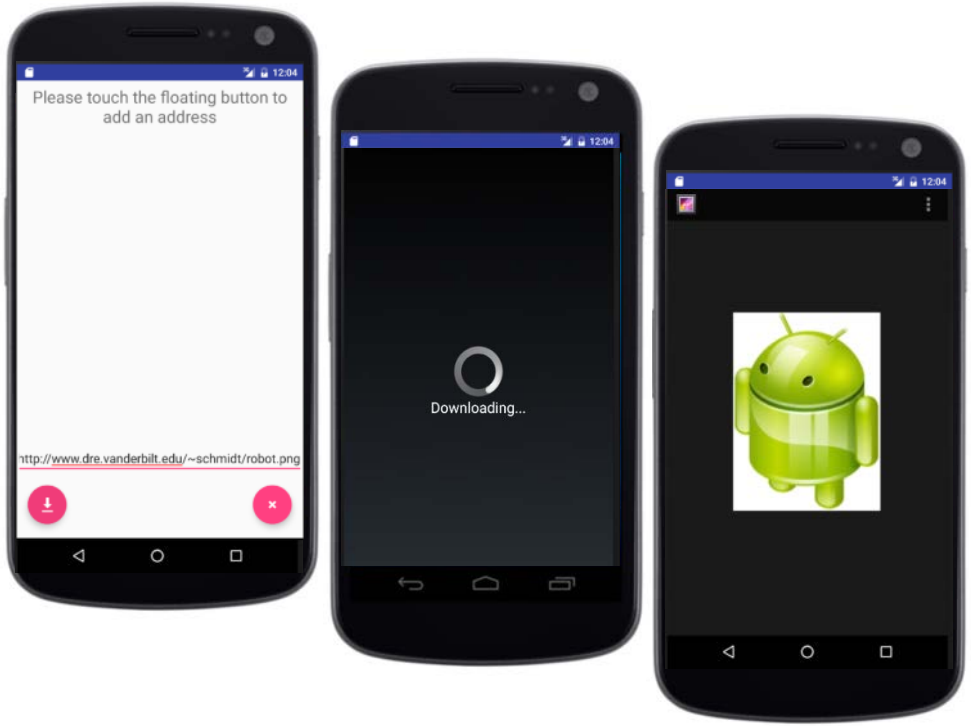
- These apps uses an intent & an activity to map the address of a contact using Android concurrency frameworks, e.g.
  - HaMeR framework
  - AsyncTask framework



We'll also examine this app's use of Content Providers in an upcoming course

Overview of ImageDownloader

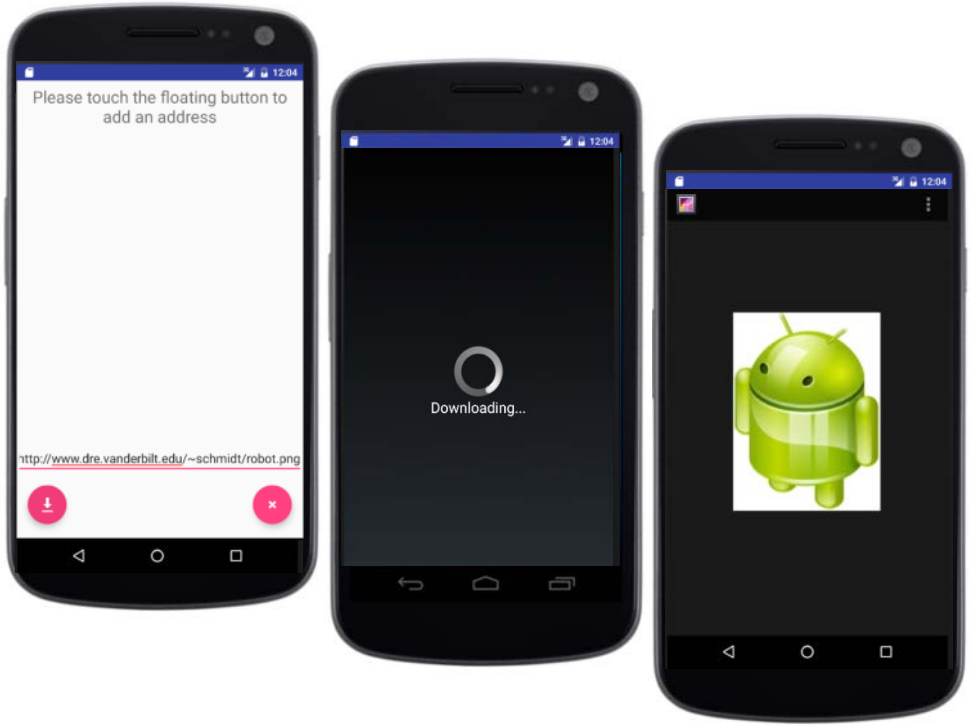
- This app uses an activity to prompt the user for a URL to an image & then uses intents & other activities to download the image & view it



See gitlab.com/vandy-aad-2/ImageDownloader

Overview of ImageDownloader

- This app uses an activity to prompt the user for a URL to an image & then uses intents & other activities to download the image & view it

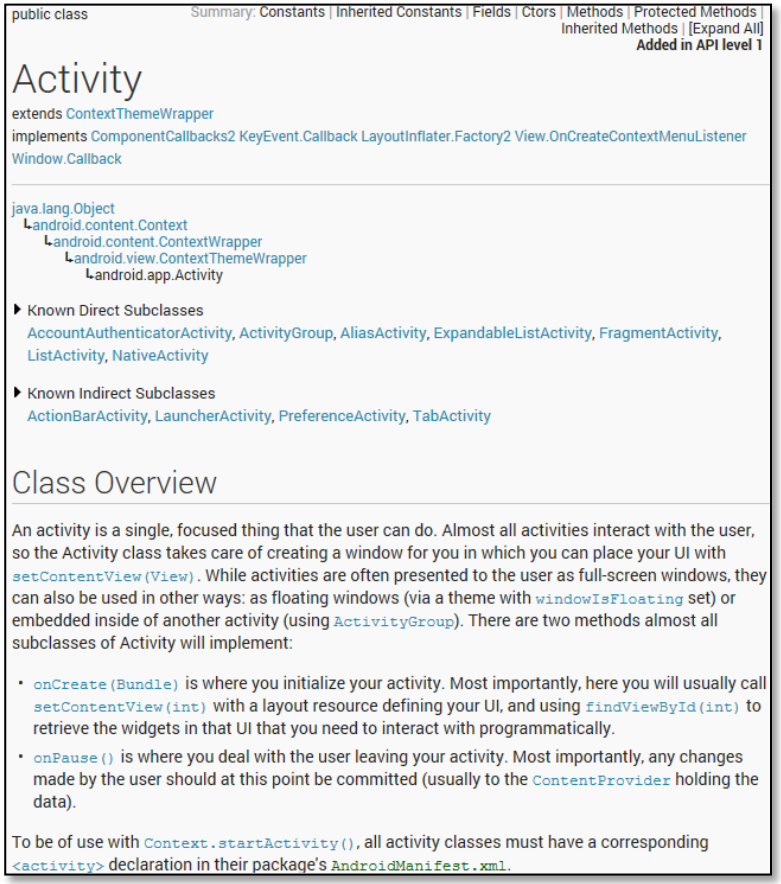


We'll also analyze variants of ImageDownloader in later courses

Steps for Implementing an Android Activity

Steps for Implementing an Android Activity

- Implementing an Activity involves multiple steps



public class — Summary: Constants | Inherited Constants | Fields | Ctors | Methods | Protected Methods | Inherited Methods | [Expand All]
Added in API level 1

Activity

extends ContextThemeWrapper
implements ComponentCallbacks2 KeyEvent.Callback LayoutInflater.Factory2 View.OnCreateContextMenuListener Window.Callback

java.lang.Object
↳android.content.Context
↳android.content.ContextWrapper
↳android.view.ContextThemeWrapper
↳android.app.Activity

▸ Known Direct Subclasses
AccountAuthenticatorActivity, ActivityGroup, AliasActivity, ExpandableListActivity, FragmentActivity, ListActivity, NativeActivity

▸ Known Indirect Subclasses
ActionBarActivity, LauncherActivity, PreferenceActivity, TabActivity

Class Overview

An activity is a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class takes care of creating a window for you in which you can place your UI with `setContentView(View)`. While activities are often presented to the user as full-screen windows, they can also be used in other ways: as floating windows (via a theme with `windowIsFloating` set) or embedded inside of another activity (using `ActivityGroup`). There are two methods almost all subclasses of Activity will implement:

- `onCreate(Bundle)` is where you initialize your activity. Most importantly, here you will usually call `setContentView(int)` with a layout resource defining your UI, and using `findViewById(int)` to retrieve the widgets in that UI that you need to interact with programmatically.
- `onPause()` is where you deal with the user leaving your activity. Most importantly, any changes made by the user should at this point be committed (usually to the `ContentProvider` holding the data).

To be of use with `Context.startActivity()`, all activity classes must have a corresponding `<activity>` declaration in their package's `AndroidManifest.xml`.

See developer.android.com/reference/android/app/Activity.html

Steps for Implementing an Android Activity

- Implementing an Activity involves multiple steps, e.g.
  - Extend the Activity class

```
public class LifecycleLoggingActivity
             extends Activity {

  ...
```

Steps for Implementing an Android Activity

- Implementing an Activity involves multiple steps, e.g.
  - Extend the Activity class

```
public class LifecycleLoggingActivity
             extends Activity {
  ...
```

Provides a common interface for interacting with a user, including operations performed when moving between lifecycle states

See www.dre.vanderbilt.edu/~schmidt/PDF/Commonality_Variability.pdf

Steps for Implementing an Android Activity

- Implementing an Activity involves multiple steps, e.g.
  - Extend the Activity class
  - Override selected lifecycle methods

```
public class LifecycleLoggingActivity
             extends Activity {
  protected void onCreate
     (Bundle savedInstanceState) {...}
  protected void onStart() {...}
  protected void onRestart() {...}
  protected void onResume() {...}
  protected void onPause() {...}
  protected void onStop() {...}
  protected void onDestroy() {...}
  ...
```

Steps for Implementing an Android Activity

- Implementing an Activity involves multiple steps, e.g.
  - Extend the Activity class
  - Override selected lifecycle methods

Subclasses can override lifecycle hook methods to do necessary work when an Activity changes state

```
public class LifecycleLoggingActivity
             extends Activity {
  protected void onCreate
     (Bundle savedInstanceState) {...}
  protected void onStart() {...}
  protected void onRestart() {...}
  protected void onResume() {...}
  protected void onPause() {...}
  protected void onStop() {...}
  protected void onDestroy() {...}
  ...
```

See www.dre.vanderbilt.edu/~schmidt/PDF/Commonality_Variability.pdf

Steps for Implementing an Android Activity

- Implementing an Activity involves multiple steps, e.g.
  - Extend the Activity class
  - Override selected lifecycle methods
  - Define other methods & nested classes needed to implement the Activity

```
public class MapLocationActivity
    extends LifecycleLoggingActivity {
  ...
  public void mapAddress(View view)
  {...}

  private void startMap() {...}

  private Intent makeMapsIntent
    (String address) { ... }

  private Intent makeBrowserIntent
      (String address) { ... } ...
```

Steps for Implementing an Android Activity

- Implementing an Activity involves multiple steps, e.g.
  - Extend the Activity class
  - Override selected lifecycle methods
  - Define other methods & nested classes needed to implement the Activity

```
public class MapLocationActivity
    extends LifecycleLoggingActivity {
  ...
  public void mapAddress(View view)
  {...}

  private void startMap() {...}

  private Intent makeMapsIntent
    (String address) { ... }

  private Intent makeBrowserIntent
      (String address) { ... } ...
```

This method is connected to a button via a layout XML file

Steps for Implementing an Android Activity

- Implementing an Activity involves multiple steps, e.g.
  - Extend the Activity class
  - Override selected lifecycle methods
  - Define other methods & nested classes needed to implement the Activity
- Update the AndroidManifest.xml file to include the Activity so Android knows about it

```
<manifest ...
   package="vandy.mooc.maplocation">
  ...
  <application
     <activity
        android:name=
        ".MapLocationActivity"
       <intent-filter> ...
       </...>
     </...>
     ...
```

If Activities aren't exported in AndroidManifest.xml they won't be accessible to other apps

See developer.android.com/guide/topics/manifest/manifest-intro.html

End of the Introduction to Android Activities (Part 1)