



# Accelerating Information Technology Innovation

<http://aiti.mit.edu>

India Summer 2012

Lecture 3 – Android User Interface (Navigation)



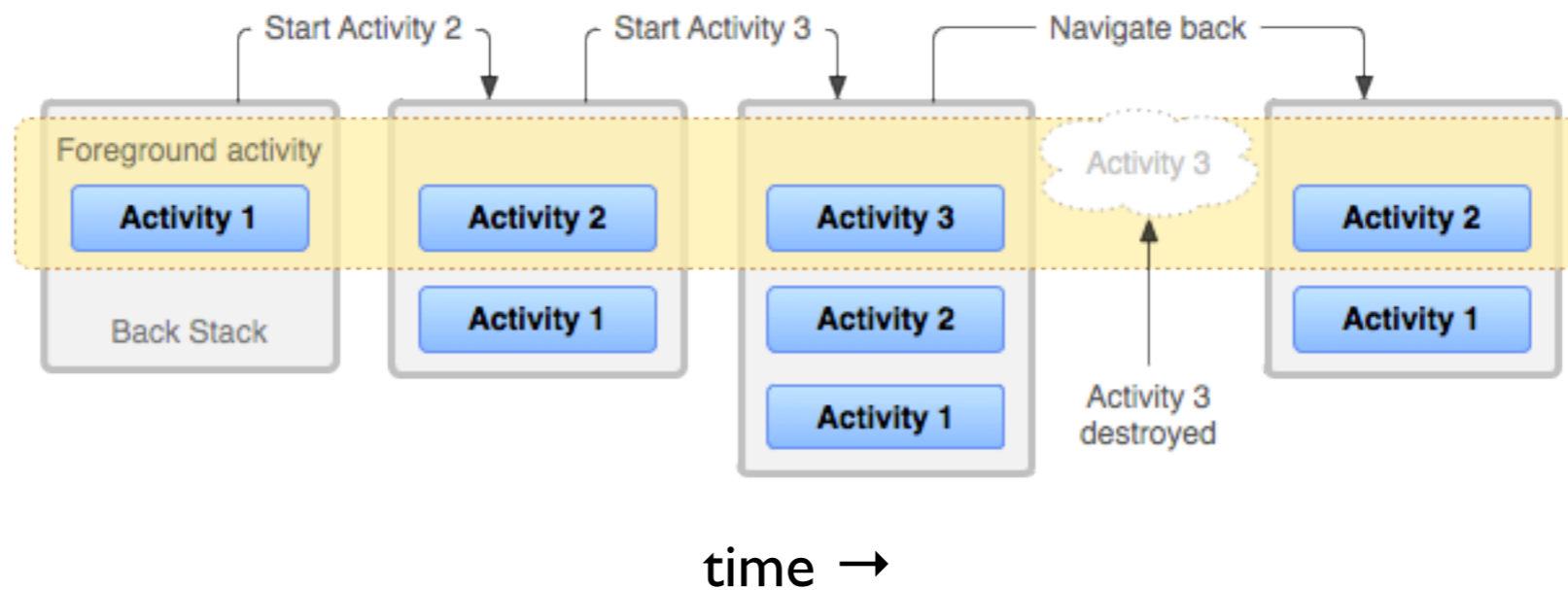
# Overview

---

- Models of app navigation
  - The activity stack
  - ListView/GridView
  - Menus
  - Dialogs and Notifications
- Using navigation in your app

# The Activity Stack

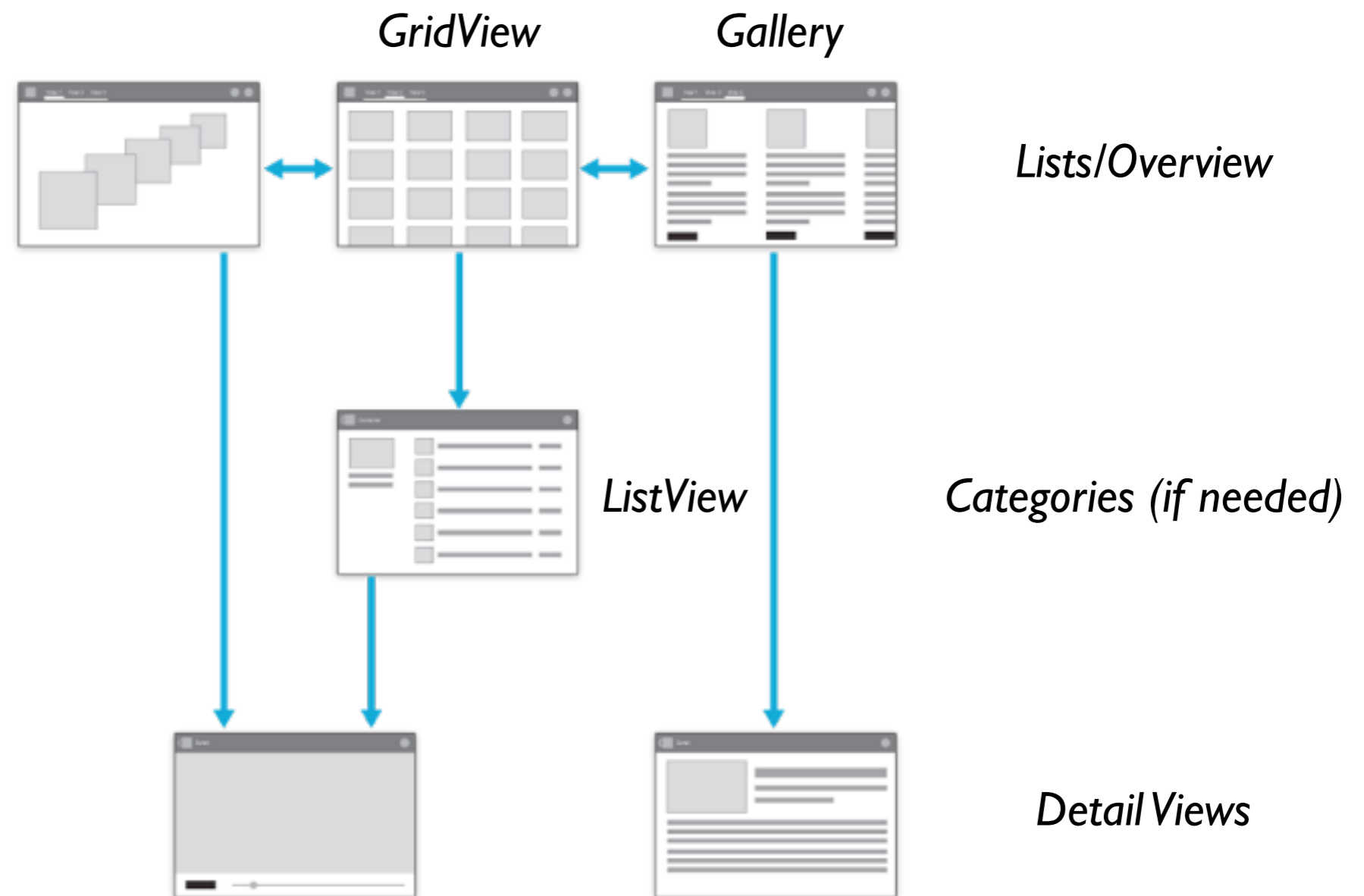
---



*The activity stack is usually used to separate **modal** actions (i.e. different modes: “reading mail” → “writing reply”)*

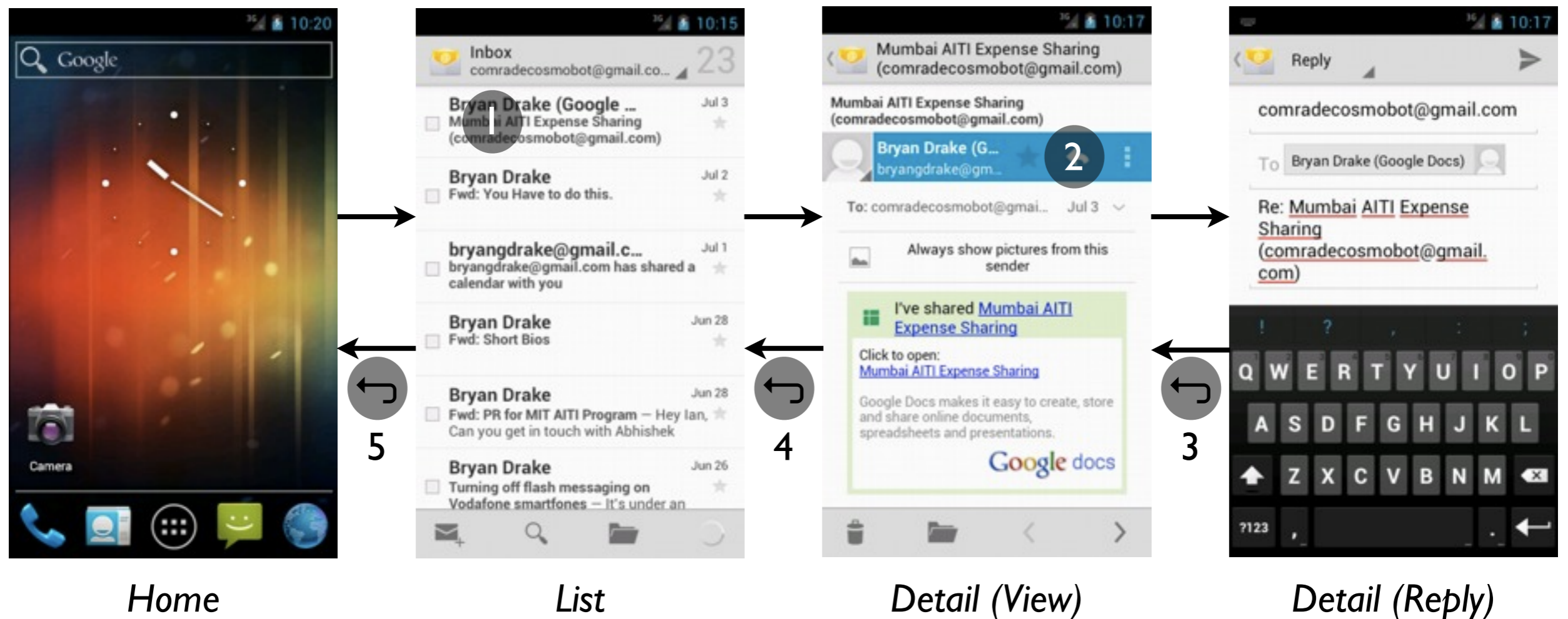
# How to use the Activity Stack

---



*Generally: Navigate from lists/overview to details!*

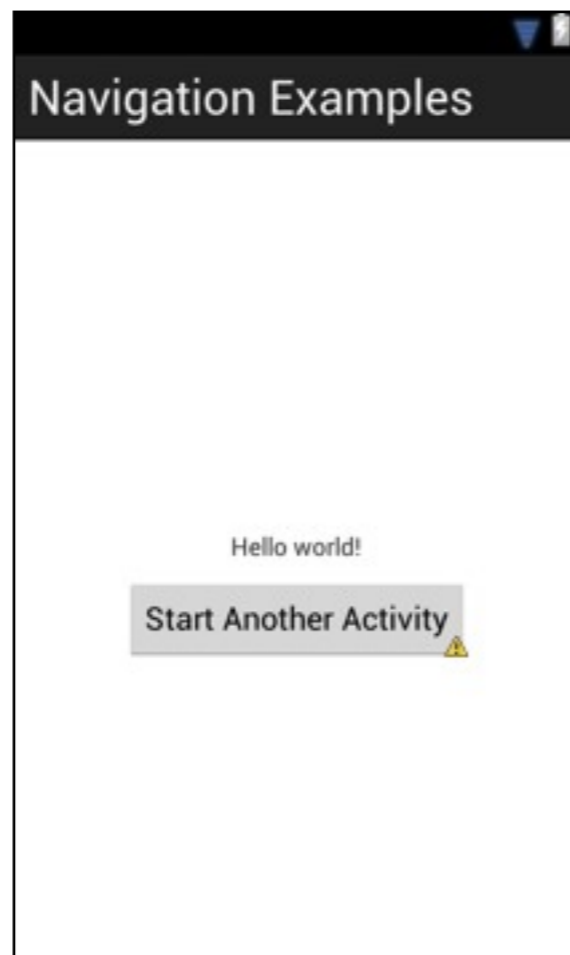
# How to use the Activity Stack



*Also useful for temporary states (e.g. replying to e-mail)*

# Using the Activity Stack

---



*NavigationExamples*



*AnotherActivity*

# The Code:

## onClickMe()

---

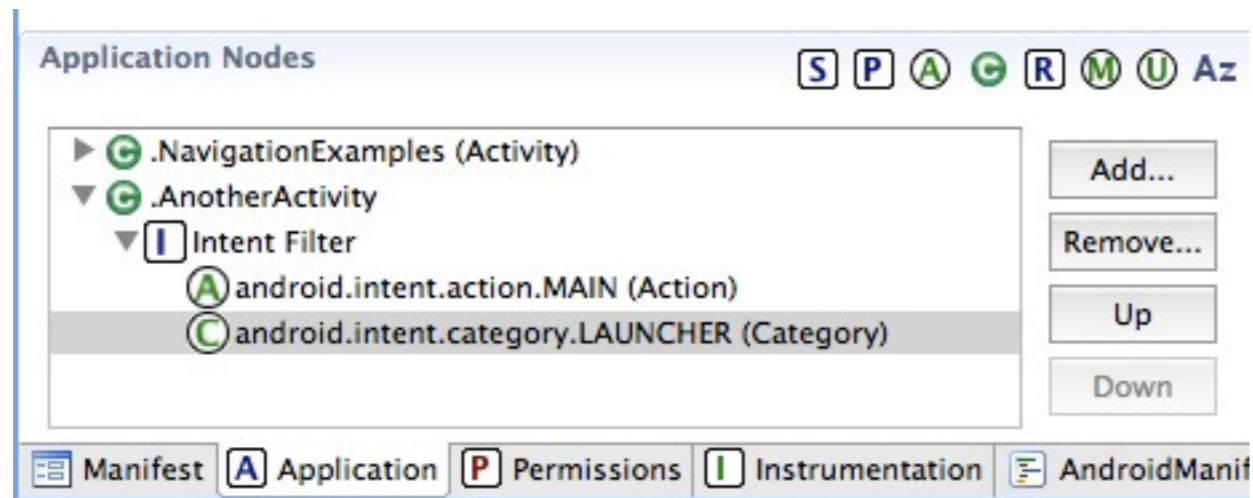
```
public class NavigationExamples extends Activity {
    public void onClickMe(View button) {
        /* Create an Intent to start AnotherActivity */
        Intent startAnotherActivity =
            new Intent(NavigationExamples.this,
                AnotherActivity.class);

        /* Start the activity. */
        startActivity(startAnotherActivity);
    }
}
```

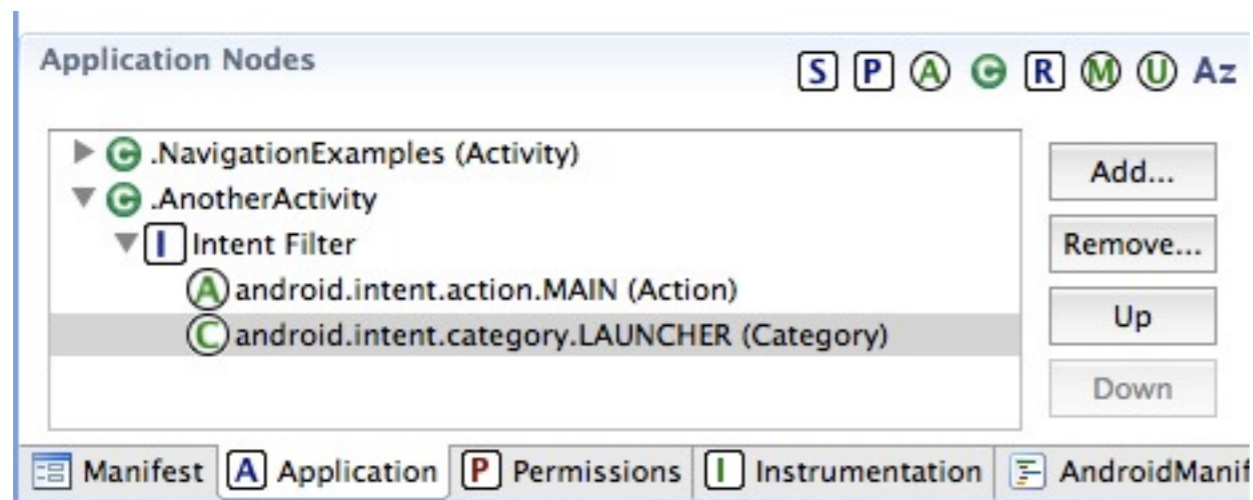
# Using the Activity Stack: Demo



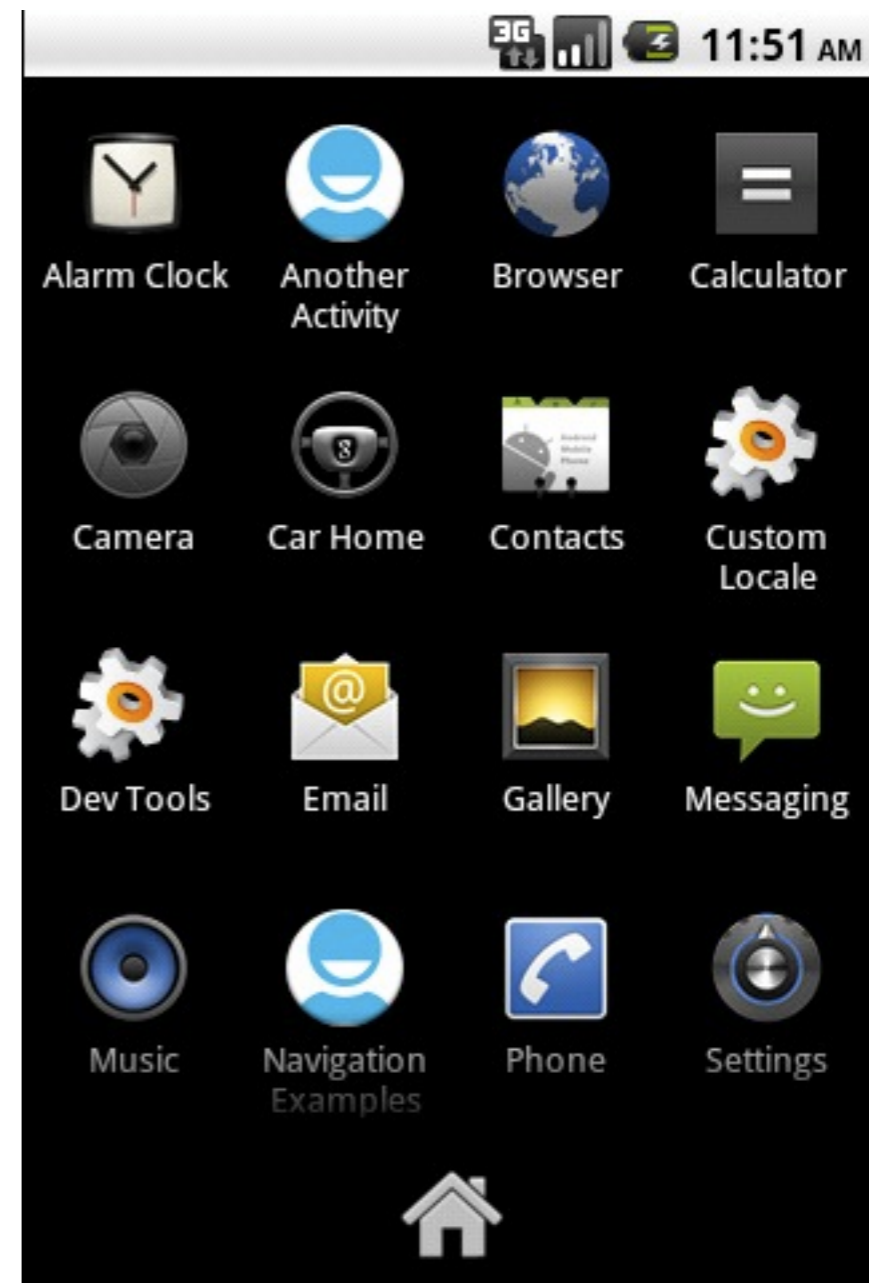
# Aside: The Manifest and Intent Filters



# Aside: The Manifest and Intent Filters



*This will appear as a separate launcher! →*



# Aside: The Manifest and Intent Filters

---



*android.intent.category.ALTERNATIVE makes it an “alternative” activity (which won’t appear in the menu)*

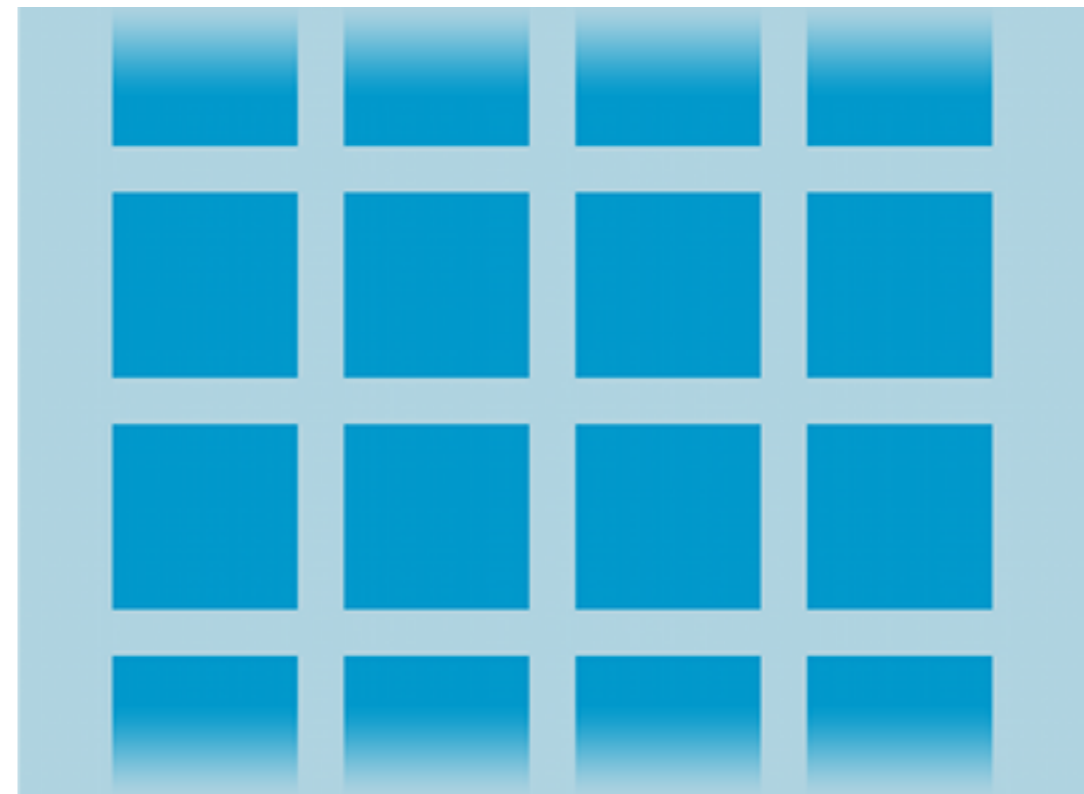
# ListView and GridView

# List View and GridView

---



*ListView*



*GridView*

*Useful to organize lists of items for further detail  
(e.g. lists of nearby restaurants, lists of train times...)*

# Adapters

---

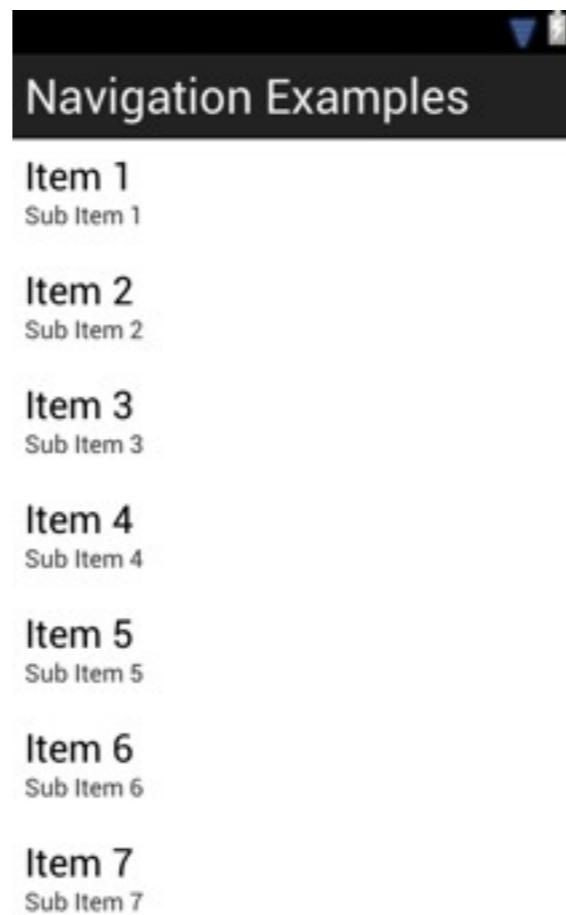


*Data Flow* →

*Adapters “wrap” a data source (database, array, etc.)  
for use in a View (ListView, GridView, Spinner, etc.)*

# Using a ListView

---



*NavigationExamples*

# The Code: onCreate()

---

```
public class NavigationExamples extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        /* Initialize the Activity */
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_navigation_examples);

        /* Set up the ListView's adapter */
        ListView myListView =
            (ListView) findViewById(R.id.myListView);
        ArrayAdapter<String> stateList =
            new ArrayAdapter<String>(this,
                android.R.layout.simple_list_item_1,
                getResources().getStringArray(R.array.states));
        myListView.setAdapter(stateList);

        /* ... */
    }
}
```



# The Code: onCreate()

---

```
/* ... */

/* Listen for clicks on each item in the ListView */
myListView.setOnItemClickListener(
    new OnItemClickListener() {
        public void onItemClick(AdapterView<?> parent,
            View view, int position, long id) {
            /* Start AnotherActivity... */
            Intent anotherActivityIntent =
                new Intent(NavigationExamples.this,
                    AnotherActivity.class);
            startActivity(anotherActivityIntent);
        }
    });
}
```

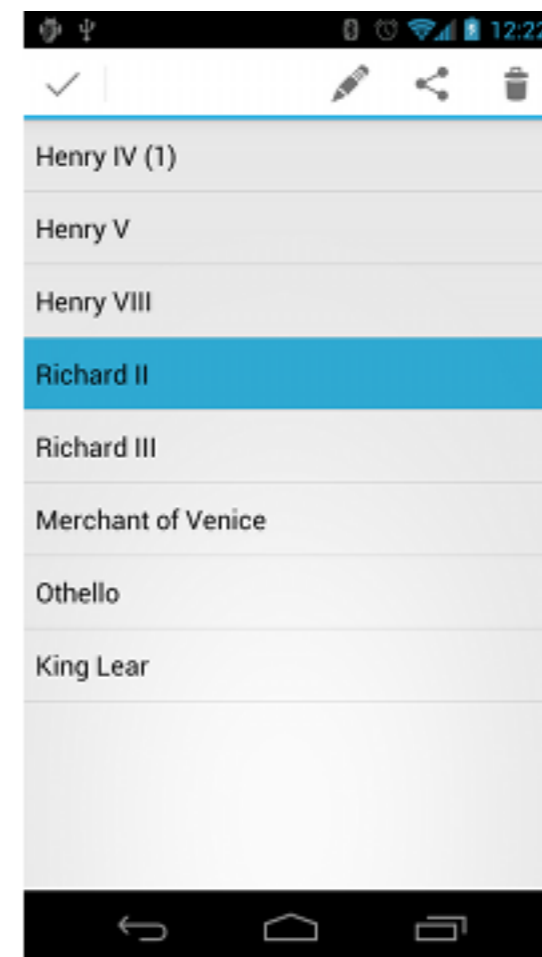
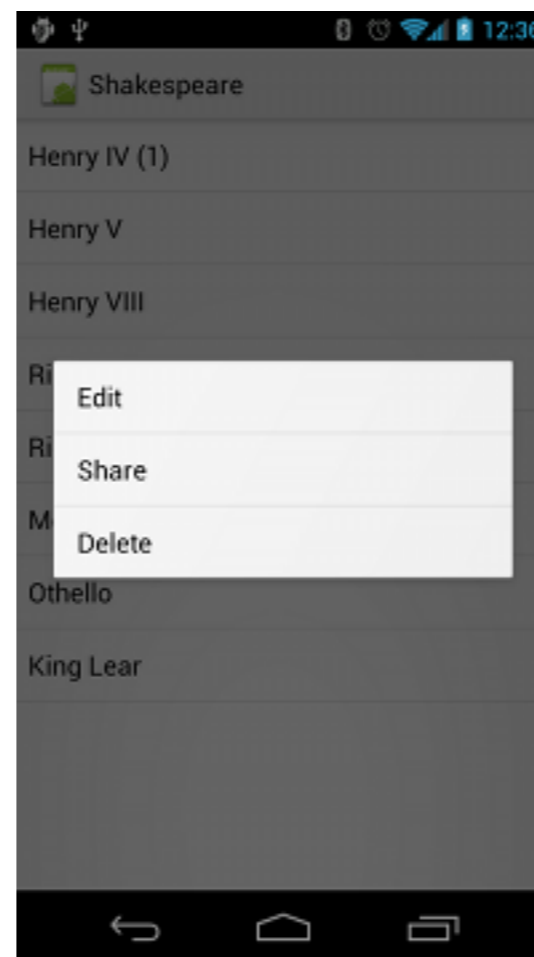
# Using a ListView: Demo

# Menus

# Menus



*Option Menu*



*Floating context menu and contextual action bar*

***Useful for common modes or actions related to an Activity  
(Context menus are useful for items IN a View)***

# Option Menu

---

*All Activities have an option menu!*

```
public class NavigationExamples extends Activity {
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(
            R.menu.activity_navigation_examples, menu);
        return true;
    }
}
```

# Option Menus

The screenshot displays the Android Studio IDE interface for editing an Android menu. The main window is titled "Android Menu" and is divided into several sections:

- Package Explorer (Left):** Shows the project structure, including folders for resources (res) and layout, and files like activity\_navigation\_example.
- Menu Elements (Center):** A list of menu items, currently containing "menu\_settings (Item)". Buttons for "Add...", "Remove...", "Up", and "Down" are visible.
- Attributes for menu\_settings (Item) (Right):** A list of attributes for the selected menu item, each with an input field and a "Browse..." button:
  - Base attributes that are available to all Item objects.
  - Id: @+id/menu\_setting
  - Menu category: (empty dropdown)
  - Order in category: 100
  - Title: @string/menu\_setti
  - Title condensed: (empty input)
  - Icon: (empty input)
  - Alphabetic shortcut: (empty input)
  - Numeric shortcut: (empty input)
  - Checkable: (empty dropdown)
  - Checked: (empty dropdown)
  - Visible: (empty dropdown)
  - Enabled: (empty dropdown)
  - On click: (empty input)
  - Show as action: never
- Bottom Toolbar:** Shows the "Layout" tab and the file "activity\_navigation\_examples.xml".

# The Code:

## onOptionsItemSelected()

---

```
public class NavigationExamples extends Activity {
    public boolean onOptionsItemSelected(MenuItem item) {
        /* Choose your action based on the item selected. */
        switch (item.getItemId()) {
            case R.id.menu_settings:
                /* Make the settings appear. */
                return true;
            default:
                return super.onOptionsItemSelected(item);
        }
    }
}
```

# Context Menu

---

- Very similar to Option Menu
- `onCreateContextMenu()` instead of `onCreateOptionsMenu()`
  - Include **super**.`onCreateContextMenu()` first
- `onContextItemSelected()` instead of `onOptionsItemSelected()`
- `((AdapterContextMenuInfo) menuItem.getContext()).id`  
gets the item context in `onOptionsItemSelected`



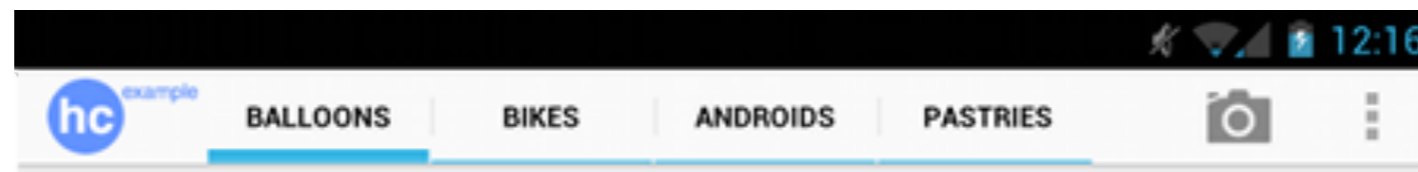
# Using Menus: Demo

**I Taught You That So I Could  
Teach You This:  
The Action Bar**

# The Action Bar

---

*At the top of the screen, this is better than a menu*



# The Action Bar

---

*At the top of the screen, this is better than a menu*



*Downside: It's only supported on Android 3.0+ (API Level 11+)*

# The Action Bar

---

*At the top of the screen, this is better than a menu*



*Downside: It's only supported on Android 3.0+ (API Level 11+)*

*But there is some sample compatibility code in*  
`<sdk>/samples/<android-version>/ActionBarCompat/`

# The Action Bar

---

***Just like Option Menus!***

- Uses `onOptionsItemSelected()` and `onOptionsItemSelected()`
- Make Action Bar Items using menu resources

# Activating the Action Bar

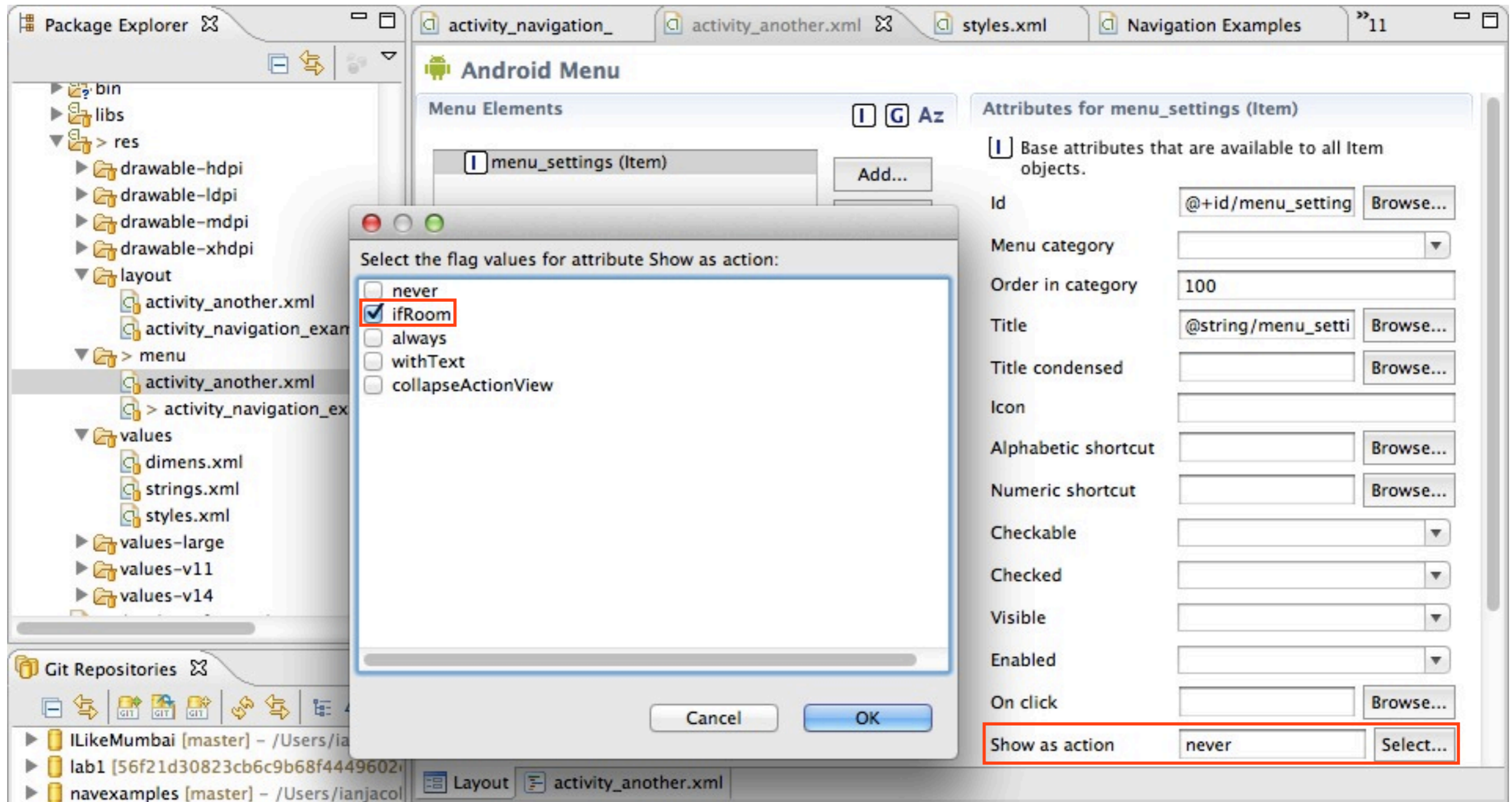
The screenshot displays the AndroidManifest.xml editor in an IDE. The left sidebar shows the project structure with 'AndroidManifest.xml' selected. The main editor area is titled 'Android Manifest' and contains the following sections:

- Manifest General Attributes:** Defines general information about the AndroidManifest.xml. Fields include Package (edu.mit.aiti.india.navexamples), Version code (1), Version name (1.0), Shared user id, Shared user label, and Install location.
- Manifest Extras:** A list of extras with 'Uses Sdk' selected. Buttons for 'Add...', 'Remove...', 'Up', and 'Down' are visible.
- Attributes for Uses Sdk:** Describes the SDK features that the containing package must be running on to operate correctly. Fields include Min SDK version (8), Target SDK version (15), and Max SDK version.

The 'Min SDK version' and 'Target SDK version' fields are highlighted with a red box. A text overlay at the bottom right of the image reads: *Min SDK or Target SDK > 11*



# Activating the Action Bar



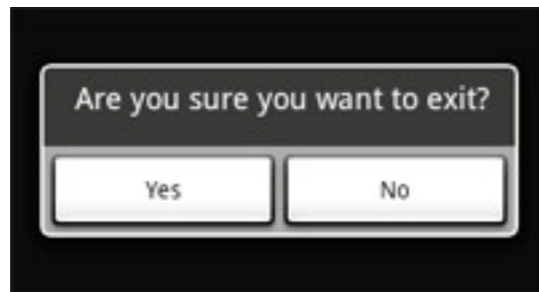


# The Action Bar: Demo

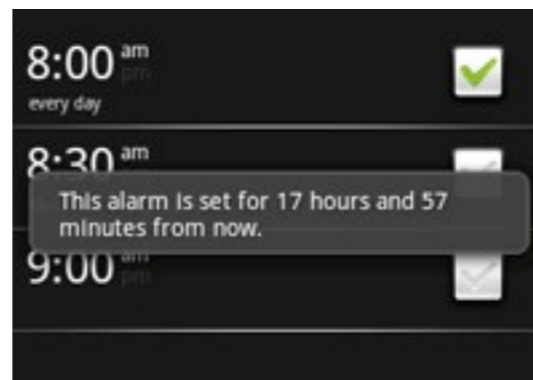
# Dialogs and Notifications

# Dialogs and Notifications

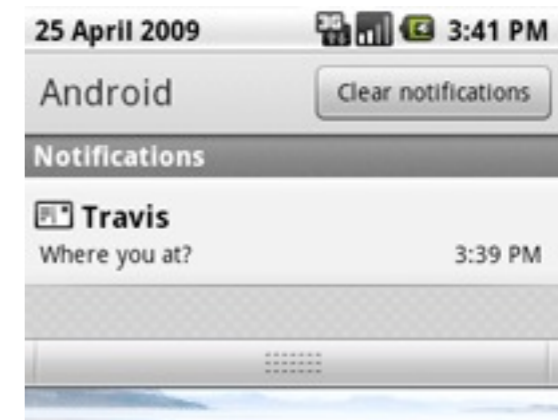
---



*AlertDialog*



*Toast Notification*



*Status Notification*

*AlertDialogs ask for simple user input.  
Other dialogs are used to show progress of an action.*

*Notifications let the user know of a background event  
(e.g. "2 new e-mails!" or "An update has been downloaded")*

# Alert Dialogs: Code

---

```
AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setMessage("You clicked the context menu!")
    .setCancelable(false)
    .setPositiveButton("Yes",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                dialog.cancel();
            }
        })
    .setNegativeButton("No",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                dialog.cancel();
            }
        });
AlertDialog dialog = builder.create();
dialog.show();
```

# Toast Notifications: Code

---

```
Toast.makeText(this,  
              "This is a toast notification!",  
              Toast.LENGTH_SHORT)  
    .show();
```

# Status Notifications: Code

---

```
/* Get the NotificationManager service. */
String ns = Context.NOTIFICATION_SERVICE;
NotificationManager mNotificationManager =
    (NotificationManager) getSystemService(ns);

/* Create the notification. */
int icon = R.drawable.ic_launcher;           // Icon
CharSequence tickerText = "The ticker text"; // Ticker Text
long when = System.currentTimeMillis();     // When to show

Notification notification = new Notification(
    icon, tickerText, when);
```

# Status Notifications: Code (Continued)

---

```
/* Prepare the event info (message) */
Context context = getApplicationContext();
CharSequence contentTitle = "My notification"; // The title
CharSequence contentText = "Hello World!"; // The text
/* This intent to restart is "delayed" for a later date. */
Intent notificationIntent = new Intent(
    this, NavigationExamples.class);
PendingIntent contentIntent = PendingIntent.getActivity(
    this, 0, notificationIntent, 0);

notification.setLatestEventInfo(
    context, contentTitle, contentText, contentIntent);

/* Send the notification. */
final int HELLO_ID = 1;
mNotificationManager.notify(HELLO_ID, notification);
```

# Resources

---

- Android Developer Site
  - “ListViews” <<http://developer.android.com/guide/topics/ui/layout/listview.html>>
  - “Menus” <<http://developer.android.com/guide/topics/ui/menus.html>>
  - “Action Bar” <<http://developer.android.com/guide/topics/ui/actionbar.html>>
  - “Dialogs” <<http://developer.android.com/guide/topics/ui/dialogs.html>>
  - “Notifications” <<http://developer.android.com/guide/topics/ui/notifiers/index.html>>
- Android Design (on the Developer Site)
  - “App Structure” <<http://developer.android.com/design/patterns/app-structure.html>>
  - “Navigation” <<http://developer.android.com/design/patterns/navigation.html>>



# More Resources

---

- Android Training (on the Developer Site)
  - “Designing Effective Navigation”  
<<http://developer.android.com/training/design-navigation/index.html>>
  - “Implementing Effective Navigation”  
<<http://developer.android.com/training/implementing-navigation/index.html>>
- Android Patterns (a list of useful interaction patterns for Android apps)  
<<http://www.androidpatterns.com/>>
- Giorgio Venturi has a presentation about Android Design Patterns  
<<http://www.closetag.com/thought-pieces/android-design-patterns/>>