Social Media Apps Programming

Developing Android Native Apps with Java (Eclipse) (MIT App Inventor)

Min-Yuh Day, Ph.D.
Assistant Professor
Department of Information Management
Tamkang University

http://mail.tku.edu.tw/myday

2013-10-17
<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Subject/Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>• 1</td>
<td>2013/09/19</td>
<td>Mid-Autumn Festival (Day off)</td>
</tr>
<tr>
<td>• 2</td>
<td>2013/09/26</td>
<td>Course Orientation and Introduction to Social Media and Mobile Apps Programming</td>
</tr>
<tr>
<td>• 3</td>
<td>2013/10/03</td>
<td>Introduction to Android / iOS Apps Programming</td>
</tr>
<tr>
<td>• 4</td>
<td>2013/10/10</td>
<td>Double Tenth Day (Day off)</td>
</tr>
<tr>
<td>• 5</td>
<td>2013/10/17</td>
<td>Developing Android Native Apps with Java (Eclipse) (MIT App Inventor)</td>
</tr>
<tr>
<td>• 6</td>
<td>2013/10/24</td>
<td>Developing iPhone / iPad Apps Native Apps with Objective-C (Xcode)</td>
</tr>
<tr>
<td>Week</td>
<td>Date</td>
<td>Subject/Topics</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>----------------------------------------------------</td>
</tr>
<tr>
<td>7</td>
<td>2013/10/31</td>
<td>Mobile Apps using HTML5/CSS3/JavaScript</td>
</tr>
<tr>
<td>8</td>
<td>2013/11/07</td>
<td>jQuery Mobile</td>
</tr>
<tr>
<td>9</td>
<td>2013/11/14</td>
<td>Create Hybrid Apps with Phonegap</td>
</tr>
<tr>
<td>10</td>
<td>2013/11/21</td>
<td>Midterm Exam Week (Midterm Project Report)</td>
</tr>
<tr>
<td>11</td>
<td>2013/11/28</td>
<td>jQuery Mobile/Phonegap</td>
</tr>
<tr>
<td>12</td>
<td>2013/12/05</td>
<td>Google App Engine</td>
</tr>
<tr>
<td>Week</td>
<td>Date</td>
<td>Subject/Topics</td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>13</td>
<td>2013/12/12</td>
<td>Google Map API</td>
</tr>
<tr>
<td>14</td>
<td>2013/12/19</td>
<td>Facebook API (Facebook JavaScript SDK) (Integrate Facebook with iOS/Android Apps)</td>
</tr>
<tr>
<td>15</td>
<td>2013/12/26</td>
<td>Twitter API</td>
</tr>
<tr>
<td>16</td>
<td>2014/01/02</td>
<td>Case Study on Social Media Apps</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Programming and Marketing in Google Play and App Store</td>
</tr>
<tr>
<td>17</td>
<td>2014/01/09</td>
<td>Final Project Presentation</td>
</tr>
<tr>
<td>18</td>
<td>2014/01/16</td>
<td>Final Exam Week (Final Project Report)</td>
</tr>
</tbody>
</table>
Outline

• Developing **Android Native Apps with Java**
  – Eclipse
  – Android Developer Tools (**ADT**) Bundle
  – Building Your First Android App

• **MIT App Inventor**
Introduction to Android /iOS Apps Programming

Native Apps

Hybrid Apps

Mobile Web Apps
Native App Development

Android - Native App Development

Native App – Interaction with Mobile Device

Android KitKat

We’re naming the next version of Android after one of our favorite chocolate treats, KitKat. Look for specially branded KitKat bars in a store near you that give you a chance to win a Nexus 7 or Google Play credit.

Learn more >
Android 1.5, Cupcake

Right from the start, Android is an open OS that can run almost any app or widget so you can do what you want to do.
Android 1.6, Donut

The world’s information is at your fingertips - search the web, get driving directions...or watch cat videos.
Android 2.0, Eclair

Make your home screen just how you want it. Arrange apps and widgets across multiple screens and in folders. Stunning live wallpapers respond to your touch.
Android 2.2, Froyo

Voice Typing lets you input text, and Voice Actions let you control your phone, just by speaking.
Android 2.3, Gingerbread

New sensors make Android great for gaming - so you can touch, tap, tilt, and play away.
Android 3.0, Honeycomb

Optimized for tablets, this release opens up new horizons wherever you are.
Android 4.0, Ice Cream Sandwich

Android comes of age with a new, refined design. Simple, beautiful and beyond smart.
Android 4.1, Jelly Bean

Android is fast and smooth with buttery graphics. With Google Now, you get just the right information at the right time.

And with more than 1 million apps on Google Play, and thousands of Android devices, you’ve got the freedom to do what you want on any device you choose.
Android 4.4, KitKat

It's our goal with Android KitKat to make an amazing Android experience available for everybody.
# Android Platform

<table>
<thead>
<tr>
<th>Android Version</th>
<th>Codename</th>
<th>API</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2</td>
<td>Froyo</td>
<td>8</td>
<td>2.2%</td>
</tr>
<tr>
<td>2.3.3-2.3.7</td>
<td>Gingerbread</td>
<td>10</td>
<td>28.5%</td>
</tr>
<tr>
<td>3.2</td>
<td>Honeycomb</td>
<td>13</td>
<td>0.1%</td>
</tr>
<tr>
<td>4.0.3-4.0.4</td>
<td>Ice Cream Sandwich</td>
<td>15</td>
<td>20.6%</td>
</tr>
<tr>
<td>4.1.x</td>
<td>Jelly Bean</td>
<td>16</td>
<td>36.5%</td>
</tr>
<tr>
<td>4.2.x</td>
<td>Jelly Bean</td>
<td>17</td>
<td>10.6%</td>
</tr>
<tr>
<td>4.3</td>
<td>Jelly Bean</td>
<td>18</td>
<td>1.5%</td>
</tr>
</tbody>
</table>

Data collected during a 7-day period ending on October 2, 2013

Android Platform

Data collected during a 7-day period ending on October 2, 2013

http://developer.android.com/about/dashboards/index.html
Android Screen Sizes and Densities

Data collected during a 7-day period ending on October 2, 2013
http://developer.android.com/about/dashboards/index.html
Android Screen Sizes and Densities

http://developer.android.com/guide/practices/screens_support.html
Android Development Environment

1. **JDK (Java Development Kit)**
   
   [Link](http://www.oracle.com/technetwork/java/javase/downloads/index.html)

2. **ADT Bundle**
   (Android Developer Tools Bundle)
   (Eclipse + ADT plugin + Android SDK + Android Platform + emulator)

   [Link](http://developer.android.com/sdk/index.html)
1. JDK (Java Development Kit)

Java SE Downloads

Java Platform, Standard Edition

Java SE 7u40
This release includes several new features like Java Mission Control, Deployment Rule Set, support for Retina display on Mac, and Hard Float ABI support on Linux ARM v7. It also includes bug fixes and enhancements. Learn more

Which Java package do I need?
- JDK: (Java Development Kit). For Java Developers. Includes a complete JRE plus tools for developing, debugging, and monitoring Java applications.
- Server JRE: (Server Java Runtime Environment) For deploying Java applications on servers. Includes tools for JVM monitoring and tools commonly required for server applications, but does not include browser integration (the Java plug-in). auto-updates, nor an installer. Learn more
- JRE: (Java Runtime Environment). Covers most end-users needs. Contains everything

Get the Android SDK

The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android.

If you're a new Android developer, we recommend you download the ADT Bundle to quickly start developing apps. It includes the essential Android SDK components and a version of the Eclipse IDE with built-in ADT (Android Developer Tools) to streamline your Android app development.

With a single download, the ADT Bundle includes everything you need to begin developing apps:

- Eclipse + ADT plugin
- Android SDK Tools
- Android Platform-tools
- The latest Android platform
- The latest Android system image for the emulator

Android Studio Early Access Preview

A new Android development environment called Android Studio, based on IntelliJ IDEA, is now available as an early access preview. For more information, see Getting Started with Android Studio.

If you prefer to use an existing version of Eclipse or another IDE, you can instead take a more customized approach to installing the Android SDK. See the following instructions:

Building Your First Android App

1. Download the Android SDK.
2. Install the ADT plugin for Eclipse
3. Download the latest SDK tools and platforms using the SDK Manager.

ADT Bundle
( Android Developer Tools Bundle )
( Eclipse + ADT plugin + Android SDK + Android Platform + emulator )

ADT Bundle

ADT (Android Developer Tools)

- With a single download, the ADT Bundle includes everything you need to begin developing apps:

  1. Eclipse + ADT plugin
  2. Android SDK Tools
  3. Android Platform-tools
  4. The latest Android platform
  5. The latest Android system image for the emulator

http://developer.android.com/training/basics/firstapp/index.html
ADT

Android Developer Tools

Build: v22.2.1-833290

This product includes Eclipse Platform, JDT, CDT, EMF, GEF, and WTP, all of which are Copyright (c) Eclipse contributors and others.

Android Developer Tools are Copyright (c) The Android Open Source Project.
Visit [http://developer.android.com](http://developer.android.com)

Installation Details
Android App Building Blocks

1. Activity
2. Service
3. Broadcast Receiver
4. Content Provider

Android App Building Blocks

1. Activity
   - Activated by an asynchronous message

2. Service

3. Broadcast Receiver

4. Content Provider

Android App Building Blocks

1. Activity
2. Service
3. Broadcast Receiver
4. Content Provider

a single screen with a user interface

Android App Building Blocks

1. Activity
2. Service
   runs in the background to perform long-running operations or to perform work for remote processes
3. Broadcast Receiver
4. Content Provider

Android App Building Blocks

1. Activity
2. Service
3. Broadcast Receiver
4. Content Provider

responds to system-wide broadcast announcements

Android App Building Blocks

1. Activity
2. Service
3. Broadcast Receiver
4. Content Provider

manages a shared set of application data

Developing Android Apps

1. **Screen Layout Design**: Views and Layouts
   - Graphical Layout
   - activity_main.xml

2. **App Components (Activity) Programming**
   - MainActivity.java
Building a Simple User Interface

• Create a Linear Layout
• Add a Text Field
• Add String Resources
• Add a Button
• Make the Input Box Fill in the Screen Width

Building a Simple User Interface

Source: http://developer.android.com/training/basics/firstapp/building-ui.html
Building a Simple User Interface

Source: http://developer.android.com/training/basics/firstapp/building-ui.html
Demo:
Building Your First Android App
with
Android Developer Tools
(ADT)
public class MainActivity extends Activity {
    public final static String EXTRA_MESSAGE = "tw.edu.tku.im.smap2013.imyday.myfirstapp"

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    /** Called when the user clicks the Send button */
    public void sendMessage(View view) {
        Intent intent = new Intent(this, DisplayMessageActivity.
            Intent.putExtra(EXTRA_MESSAGE, message);
        startActivity(intent);
    }
}
Welcome to Android application development!

This class teaches you how to build your first Android app. You'll learn how to create an Android project and run a debuggable version of the app. You'll also learn some fundamentals of Android app design, including how to build a simple user interface and handle user input.

Before you start this class, be sure you have your development environment set up. You need to:

1. Download the Android SDK.
2. Install the ADT plugin for Eclipse (if you'll use the Eclipse IDE).
3. Download the latest SDK tools and platforms using the SDK Manager.

If you haven't already done these tasks, start by downloading the [Android SDK](http://developer.android.com/training/basics/fundamentals/index.html) and following the install steps. Once you've finished the setup, you're ready to begin this class.

This class uses a tutorial format that incrementally builds a small Android app that teaches you some fundamental concepts about Android development, so it's important that you follow each step.

Start the first lesson »
Creating an Android Project

An Android project contains all the files that comprise the source code for your Android app. The Android SDK tools make it easy to start a new Android project with a set of default project directories and files.

This lesson shows how to create a new project either using Eclipse (with the ADT plugin) or using the SDK tools from a command line.

**Note:** You should already have the Android SDK installed, and if you’re using Eclipse, you should also have the ADT plugin installed (version 21.0.0 or higher). If you don’t have these, follow the guide to [Installing the Android SDK](http://developer.android.com/get-started/前/post) before you start this lesson.

Create a Project with Eclipse

1. Click **New** in the toolbar.
2. In the window that appears, open the Android folder, select Android Application Project, and click **Next**.
3. Fill in the form that appears:
   - **Application Name** is the app name that appears to users. For this project, use “My First App.”
   - **Project Name** is the name of your project directory and the name visible in Eclipse.

http://developer.android.com/training/basics/firstapp/creating-project.html
New Android Application

The prefix 'com.example.' is meant as a placeholder and should not be used

Application Name: My First App
Project Name: MyFirstApp
Package Name: com.example.myfirstapp

Minimum Required SDK: API 18: Android 4.3 (Jelly Bean)
Target SDK: API 18: Android 4.3 (Jelly Bean)
Compile With: API 18: Android 4.3 (Jelly Bean)
Theme: Holo Light with Dark Action Bar

The application name is shown in the Play Store, as well as in the Manage Application list in Settings.

< Back  Next >  Cancel  Finish
New Android Application

Creates a new Android Application

Application Name: My First App
Project Name: MyFirstApp
Package Name: tw.edu.tku.im.smap2013.imyday.myfirstapp

Minimum Required SDK: API 8: Android 2.2 (Froyo)
Target SDK: API 18: Android 4.3 (Jelly Bean)
Compile With: API 18: Android 4.3 (Jelly Bean)
Theme: Holo Light with Dark Action Bar

The package name must be a unique identifier for your application. It is typically not shown to users, but it "must" stay the same for the lifetime of your application; it is how multiple versions of the same application are considered the "same app". This is typically the reverse domain name of your organization plus one or more application identifiers, and it must be a valid Java package name.

Welcome!

The Android Developer Tools provide a first-class dev integrated development environment to build apps an immediately begin developing apps and test them on an emulator. Below, follow the steps to build your first app.

Build Your First App
If you're new to Android, follow the instructions to install an emulator and setup your development environment.

Design Your App
Before you begin developing your app, you may want to design the layout and user interface.

Test Your App
The Android framework provides tools to test and debug your app. You can also use an emulator to run your app and test it.

Tutorials

1. Build Your First App
2. Design Your App
3. Test Your App

Android IDE

Welcome!
New Android Application

Creates a new Android Application

Application Name: My First App
Project Name: MyFirstApp
Package Name: tw.edu.tku.im.cmap2013.imyday.myfirstapp
Minimum Required SDK: API 8: Android 2.2 (Troyo)
Target SDK: None
Compile With: Holo Dark
Theme: Holo Light with Dark Action Bar

The package name must be a unique identifier for your application. It is typically not shown to users, but it "must" stay the same for the lifetime of your application; it is how multiple versions of the same application are considered the "same app". This is typically the reverse domain name of your organization plus one or more application identifiers, and it must be a valid Java package name.
Create Activity

Select whether to create an activity, and if so, what kind of activity.

- Create Activity
- Blank Activity
- Fullscreen Activity
- Master/Detail Flow

Blank Activity

Creates a new blank activity, with an action bar and optional navigational elements such as tabs or horizontal swipe.
Blank Activity
Creates a new blank activity, with an action bar and optional navigational elements such as tabs or horizontal swipe.

Activity Name: MainActivity
Layout Name: activity_main
Navigation Type: None

The name of the activity class to create

Finish
package tw.edu.tku.im.smap2013.imyday.myfirstapp;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

}
package tw.edu.tku.im.smap2013.imyday.myfirstapp;

import android.os.Bundle;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
public class MainActivity extends Activity {

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}

@Override
public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);
}

public void onPrepareOptionsMenu(Menu menu) {
    super.onPrepareOptionsMenu(menu);
}

public void onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    menu.add(0, R.id.menu_item_1, 1, "Menu Item 1");
    menu.add(0, R.id.menu_item_2, 2, "Menu Item 2");
    menu.add(0, R.id.menu_item_3, 3, "Menu Item 3");
}

public void onOptionsItemSelected(MenuItem item) {
    super.onOptionsItemSelected(item);
}

No compatible targets were found. Do you wish to add a new Android Virtual Device?

No  Yes
on the second activity.

Figure 2. Both activities in the final app, running on Android 4.0.

That's it, you've built your first Android app!

To learn more, follow the link below to the next class.

http://developer.android.com/training/basics/firstapp/starting-activity.html
public class MainActivity extends Activity {
    public final static String EXTRA_MESSAGE = "tw.edu.tku.im.smap2013.imyday.myFirstApp";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    /** Called when the user clicks the Send button */
    public void sendMessage(View view) {
        Intent intent = new Intent(this, DisplayMessageActivity.class);
        EditText editText = (EditText) findViewById(R.id.editTextMessage);
        String message = editText.getText().toString();
        intent.putExtra(EXTRA_MESSAGE, message);
        startActivity(intent);
    }
}
Hello World

Myday
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="horizontal">
    <EditText android:id="@+id/edit_message"
        android:layout_weight="1"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="@string/edit_message" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_send"
        android:onClick="sendMessage" />
</LinearLayout>
<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: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=".DisplayMessageActivity" >

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

</RelativeLayout>
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="app_name">My First App</string>
  <string name="action_settings">Settings</string>
  <string name="hello_world">Hello world!</string>
  <string name="edit_message">Enter a message</string>
  <string name="button_send">Send</string>
  <string name="title_activity_main">MainActivity</string>
  <string name="title_activity_display_message">My Message</string>
</resources>
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="tw.edu.tku.im.smap2013.imyday.myfirstapp"
    android:versionCode="1"
    android:versionName="1.0"/>

<uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="18"/>

<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme">
    <activity
        android:name="tw.edu.tku.im.smap2013.imyday.myfirstapp.MainActivity"
        android:label="@string/app_name">
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
    </activity>
    <activity
        android:name="tw.edu.tku.im.smap2013.imyday.myfirstapp.DisplayMessageActivity"
        android:label="@string/title_activity_display_message"
        android:parentActivityName="tw.edu.tku.im.smap2013.imyday.myfirstapp">
        <meta-data
            android:name="android.support.PARENT_ACTIVITY"
            android:value="tw.edu.tku.im.smap2013.imyday.myfirstapp"/>
    </activity>
</application>
</manifest>
package tw.edu.tku.im.smap2013.imyday.myfirstapp;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.MenuItem;
import android.support.v4.app.NavUtils;
import android.annotation.SuppressLint;
import android.annotation.TargetApi;
import android.os.Build;
import android.content.Intent;
import android.widget.TextView;

public class DisplayMessageActivity extends Activity {

    @SuppressLint("NewApi")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_display_message);
        // Show the Up button in the action bar.
        setupActionBar();

        // Get the message from the intent
        Intent intent = getIntent();
        String message = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);

        // Great the text view
        TextView textView = new TextView(this);
        textView.setTextSize(40);
        textView.setText(message);

        setContentView(textView);
    }

    // ...
package tw.edu.tku.im.smap2013.imyday.myfirstapp;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.content.Intent;
import android.widget.EditText;

public class MainActivity extends Activity {
    public final static String EXTRA_MESSAGE = "tw.edu.tku.im.smap2013.imyday.myfirstapp.MESSAGE";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    /**
     * Called when the user clicks the Send button */
    public void sendMessage(View view) {
        Intent intent = new Intent(this, DisplayMessageActivity.class);
        EditText editText = (EditText) findViewById(R.id.edit_message);
        String message = editText.getText().toString();
        intent.putExtra(EXTRA_MESSAGE, message);
        startActivity(intent);
    }
}
/**
 * Set up the {@link android.app.ActionBar}, if the API is available.
 */
@TargetApi(Build.VERSION_CODES.HONEYCOMB)
private void setupActionBar() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
        getActionBar().setDisplayHomeAsUpEnabled(true);
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.display_message, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
    case android.R.id.home:
        // This ID represents the Home or Up button. In the case of this
        // activity, the Up button is shown. Use NavUtils to allow users
        // to navigate up one level in the application structure. For
        // more details, see the Navigation pattern on Android Design:
        //
        // http://developer.android.com/design/patterns/navigation.html#up-vs-back
        //
        NavUtils.navigateUpFromSameTask(this);
        return true;
    }
    return super.onOptionsItemSelected(item);
}
Alternatives for Developing Android Apps

• MIT App Inventor
  – http://appinventor.mit.edu/

• Appery.io
  – develop apps for Android (iOS / Windows Phone).
  – http://appery.io

• Appnotch
  – drag-and-drop service that allows you to develop apps for Android (iOS).
  – http://www.appnotch.com/
Welcome to MIT App Inventor

Teach
Educator Resources

Explore
Information & Tutorials

Invent
Create Mobile Apps

© 2012 Massachusetts Institute of Technology
Contact Us

http://appinventor.mit.edu/
Accelerate Mobile Innovation in the Enterprise

The only cloud-based platform with visual development tools and integrated backend services

Try Our Free Plan

Featured by

Stay tuned for news about our joint webinar with Heroku next month by checking our plancast page.

http://appery.io/
Easiest way to create Apps.

✓ Build HTML5, iOS and Android apps.
✓ Just drag & drop. No coding required.
✓ Nothing to install. Build on Cloud.
✓ Get your live app in 3 easy steps.

Start Building for Free

Reasons AppNotch is getting popular

Transform your Ideas to Apps!
Build interactive mockups, wireframes and clickable working prototypes in HTML5, Android, iPhone and iPad.

Start Prototyping

Beautiful, Flexible and Powerful!
Create topnotch apps with drag & drop templates, themes, widgets, controls, forms with instant hosting.

Start Exploring

Personal Tour and Private Demo!
We'll show you an addictively easy way to create beautiful web, native and hybrid apps, without coding.

Schedule a Demo

http://www.appnotch.com/
References

• Android Developer: http://developer.android.com/
• MIT App Inventor: http://appinventor.mit.edu/
• Native, Web or Hybrid Mobile Apps?, https://www.youtube.com/watch?v=Ns-JS4amlTc