Assignment 3

Help

Assignment 3: "My first steps as an Android Developer"

On this page:

Overview App Ideas Guidelines Rules Submission Checklist Grading FAQ

Overview

For this third and last assignment you will create **1 app** using at least 4 of the concepts and ideas presented in Week 6 and Week 7. In short, you can use your Android knowledge of:

- ScrollView
- LinearLayout
- TextView
- EditText
- TextWatcher
- Sharing text
- Visibility
- Simple animation
- Preferences
- Intents
- ImageView
- OnClickListener
- PostDelayed
- Runnable
- Bitmaps
- Mutable Bitmaps
- Canvas
- Saving and Sharing images

You will also likely need some Java knowledge including:

- String methods
- String
- View and int fields
- Using "if" "while"
- Creating new Views and other objects
- Catching exceptions
- Working with streams and Files

- Implementing interfaces

This assignment allows for more flexibility and creativity than before. As a beginner, we encourage you to start simply and stay close to the lecture examples. You may copy-paste small portions of code from Lawrence's examples but the complete app should be original and written for the purposes of this assignment so that it demonstrates your own ability to write an Android program. You may not copy entire examples from this course or any other source—that would be against the Coursera Honor Code and grounds for removal from the course. Furthermore, you would be cheating yourself out of the real-life opportunity to have someone else give you helpful feedback on your code!

We have included some suggestions on the kind of apps that would earn full marks. After the deadline for submitting these apps has passed, you will be asked to evaluate the apps of some of your peers using the grading rubric provided. Some of your peers will be evaluating your app at that time. When the peer evaluation period is over, you will be able to see the feedback provided by your peers.

- Due: See the Week 7 Overview page for details
- Grading: This activity is worth 10 points or 40% toward your badge from Illinois. It is calculated based upon the scores awarded by your peers. In the final week of the course, you will be asked to evaluate some of your peers' submissions. Failure to do so will result in a 20% penalty applied to your submission.
- Your app does not need to be completely bug free but the main functionality should work and any issues should be documented.
- Have questions about this assignment? Need some help? Want to offer a helping hand to others? Discuss this assignment here!

Requirements:

- For privacy, the package name should **not** include your real name. Remember to use the format abc.def (lowercase words, separated with periods).
- The button text, app icon, and name are up to you but should represent the purpose and contents of the app respectively in some simple way.
- The minimum SDK value will be 10.
- Add an email account (we suggest creating and using a throw-away account) to your emulator so that you can test email intents.

App Ideas

The outlines here are just **suggestions** of potential apps that you could create for full points. If you do not have any creative ideas yourself, you can use these as a starting point.

App Idea #1 "Fill-in-the-blanks"

> "My app demonstrates use of: ScrollView, LinearLayout, Animation, EditText, TextView, changing visibility, and using try-catch blocks, Strings, and Intents. My app combines the user entered text with some pre-made text to create an amusing story which can then be shared by SMS (text) message. Each EditText has a helpful text hint (e.g. "Place", "Action (verb)", "Famous person", etc.). There is also a TextView to display the completed story and two buttons.

> When the first button is clicked the story is generated, the EditTexts are made invisible, and a second button is made visible. When the second button is clicked the story is shared as an SMS message. I used a try-catch block so that if starting the SMS intent fails, the story is displayed as a toast message instead.
>
> Known issues/Future work: I've only tested this on the emulator (ARM-10). I want to shorten the text if it's more than 160 characters."

Some advanced suggestions

If you've programmed before you may enjoy employing one or more of these advanced ideas to improve the user experience in some way:

- Add an additional email button or checkbox that allows the user to share the story by email instead (and does not cause the app to crash if starting the activity fails!).
- Add an autofill button that sets the EditText entries to some words chosen randomly each time and stored in a string-array resource.
- Add a reset button that resets the app back to the original state.
- Use TextWatcher to make later EditTexts appear only once the user has entered at least one character into the earlier EditTexts.

App Idea #2 "That Meme was mine!"

> "My app demonstrates the use of: picking an external image, ImageViews, Canvas, creating a mutable bitmap, drawing text and a bitmap, and using preferences. My app will modify and share pictures, but the shared pictures include a tiny graphical stamp so that they can be traced back to myself and a given time too. I can prove that they were once my photos because I also have the original. I used preferences to ensure that each modified picture has a unique value printed on it."

Some advanced suggestions

If you've programmed before you may enjoy employing one or more of these advanced ideas to improve the user experience in some way:

- Make the stamp partially transparent and include another bitmap, today's date and time (in any reasonable format) and an integer value.
- Use Paint's "measureText" method to find out the size of your message before you paint it.

App Idea #3 "Programming Art?"

> "My app demonstrates the use of: Canvas, creating a mutable bitmap, drawing text and a bitmap, and using preferences. I used a mutable bitmap and the canvas to create brand new artwork every time you run it! If you like the picture you can choose to share it with your friends. I used a while loop to draw 100 lines of different colors and a whole load of text. I discovered you can rotate and scale things on the canvas to generate some pretty cool effects. I used Runnable and postDelayed so that the app automatically closes after 10 seconds"

App Idea #4 "A picture for my beloved"

> "I'm a beginner so my app is pretty simple but it still demonstrates basic use of: ImageView, Animation, working with interfaces (ClickListener and OnLongClickListener), and email intent. Plus this app has a use—I'm giving this to someone for Valentine's Day! I created an ImageView in Java and then set it to display a png file in my resources. I experimented with Runnable and postDelayed, OnClick and onLongClick so that the app changes the image when you click. Try it! The app uses an animation and then displays the second picture. I used postDelayed and Runnable to finish the app after 2 seconds. If you long-click it displays the email client with my email address filled in (I changed it for submission) so that my Valentine can email me anytime they want."

Guidelines

Step 1: Create Your App

Be sure to test your app in an API level 10 emulator. Other than that, the content of the app is entirely up to you to decide! You will need 1 screenshot, your Activity Java code from your activity file(s), and a signed apk file for submission.

- Your app must work on the API Level 10 emulator.
- You app must demonstrate use of a subset (**at least 4**) of the Android programming topics introduced in Week 6 and/or Week 7 .
- Your app should be simple but interesting, fun or useful.
- Just one java file! Your app's code should be entirely within your Activity.java file and may not use third-party libraries (other than Android's support library).
- Your app code cannot be too large – less than 300 lines (ideally 20 – 200 lines). Hint: Eclipse status line tells you your current line position.
- Your code should include comments that explains how the main functionality of your app is implemented.

Step 2: Refine and Test Your App

Check that your app runs correctly on the emulator API-level 10.

Be sure to add a throw-away email account to your emulator so that you can test email intents.

Step 3: Capture Screenshot to Submit

Use the screenshot functionality in the Eclipse Devices view to take 1 screen capture picture directly from the running device or emulator. Do not use your Windows/Mac/Linux screen capture tool. Your screen capture can be of any size and any density.

Step 4: Export Your App

Export your app as a signed apk file. The apk file should be valid for at least 50 years. Hint: You will need to create a key and keystore if you have not already done so.

Rules

- The app must be brand new, created from scratch, and created purely for this assignment for this particular instance of this course. You may not reuse layouts or text from existing Android projects, but you may copy portions of code from Lawrence's examples. However, the purpose here is to prove to yourself and to your peer-graders that you really can create a new app and competently use the topics from Week 6 and/or Week 7.
- The app's manifest file only may include INTERNET permission and write WRITE_EXTERNAL_STORAGE. No other permissions are allowed.
- he app may only perform its stated or specified purpose. The app must not perform anything malicious or hide any devious actions.
- The app should meet reasonable expectations of decency to a worldwide audience. To use a movie analogy, in the United States, Australia, New Zealand, and possibly others, this would be a "G" or "PG" rating; in the United Kingdom, a “U” or “PG” rating; in India, a “U” rating. We're not stopping you from making other kinds of apps—just don't submit them as an assignment!
- Your apps must at least work on emulator level 10. To have confidence in your expected score, check to make sure that your app works on an emulator API level 10 and other higher-level emulators or devices if you have them.

Submission Checklist

Check that you've completed all of the above steps and then gather the following materials to have ready to upload at the link below:

- The signed apk file.
- A screenshot of your app.
- Briefly answer the following 3 questions:
  - What does your app do?
  - Why did you decide to build the app?
  - What do you remember most about your development experience? For example, what was the hardest part of this assignment or the part that required the most time?
- Your complete Java file of your Activity. For submission, you can post the contents of your Java file directly into the text box.

Submit App for Peer Evaluation

Grading

Grading Questions

Your peers will grade your app against each of the following criteria:

- Does the app require any permissions other than INTERNET and/or WRITE_EXTERNAL_STORAGE to install? If yes: Do not continue. The review stops immediately

and the project is awarded 0 points.
- Does the app implement at least 4 topics introduced in Week 6 or Week 7? (see the beginning of this assignment for a list of topics).
- Did the user upload a screenshot?
- Was an apk and the activity Java file included?
- Did the participant answer the 3 questions?
- For open feedback: Is the code commented? Can you find comments that explain how or why the code works? Do you have any advice on how to improve the app?

Rubric

You can earn up to 10 points for your app. Your peers will award points based upon the following scale:

- **10 out of 10 points—Excellent:** The assignment is fully completed: The student demonstrated a competency and used many (at least 4) topics introduced in Week 6 and/or Week 7. No more than 1 error or omission was found.
- **8 out of 10 points—Sufficient:** The assignment is almost complete. At least 2 errors or significant omissions were found, but these were simple oversights and could be fixed fairly quickly.
- **5 out of 10 points—Partial:** The assignment was partially completed but had significant errors and would still require significant time to finish.
- **0 out of 10 points—Poor:** The assignment fell far short of being complete or sufficient.

Feedback

When reviewing the work submitted by a fellow peer, install and play with the app they provide. Give the app developer some constructive advice and some helpful feedback on their efforts. Please congratulate them and give them some feedback or comments about their code.

FAQ

Q. What if I have more questions?
A. Use the forum. Don't Panic! This assignment is meant to be educational (by helping you learn about computer science), to give you a sense of accomplishment (by having you create a specific app using everything you learned), and to allow you to be creative (in that you made your own app!).

Q. Can my app do more than what is specified?
A. Yes but review the grading rubric. Be sure to test your app carefully.

Q. Can I publish my completed app (e.g., on a website or in the Google Play Store)?
A. Yes, but we ask that you wait until after this assignment is over.

Q. Can I include text, images, or audio from site XYZ?
A. Yes, provided you are complying with the license and copyright information of the materials. For example, you may need to include any attribution information in your text (e.g., "Text from Wikipedia"). Here is a list of many Creative Commons-licensed and Public Domain images and sound

files that often can be used if given proper attribution. In the complete project this attribution might belong in a README or about dialog. For grading purposes, a simple comment in your Java file is acceptable.

Q. How I do upload a Java file if Coursera does not accept a Java file?
A. Copy-Paste the contents of your file. You may find the <code> tag can be used to format your code. Alternatively rename your file to end with a plain text

Q. Can I submit my existing app?
A. No. You need to write it from scratch.

Q. Must I export my app as an apk using a real key in a key store, or can I use the one I found inside my project's bin/directory?
A. The bin/apk file is signed with your local debug key—you cannot upload that one. You need to follow the instructions and export your project as a signed Android apk file and use your own key.

Q. Can my app use XYZ android component/feature?
A. Yes, as long as your app runs in the emulator API level 10 and does not specify any additional permissions. We strongly suggest you build something simple first—get your assignment finished and then try harder stuff!

Q. Can my second app use libGDL/web framework/XYZ non-Android technology?
A. No. The purpose of this assignment is to work with Android directly, not a framework that hides or wraps Android.

Q. But what about XYZ? I'm scared to start.
A. Just jump in—it's time to get your hands dirty! Try building these apps and see how far you can get before you get stuck. Enjoy the process. Expect to get stuck. Expect to help each other in the forums. Expect to celebrate when your app works!

Q. Can I get help on the forums/on website XYZ?
A. You can get help when things go wrong, but the app development must be yours. Show the world that you can create a new app from scratch.

Q. Can I pair-program/work with someone else?
A. Feel free to give help and get help but this app should be your own work.

Created Tue 28 Jan 2014 6:46 AM PST

Last Modified Tue 4 Feb 2014 7:28 AM PST