QuickPic App

Zak Olyarnik
Dana Thompson
Harshil Patel
Sam Caulker

Project Description

Instant picture uploads to a private online directory:

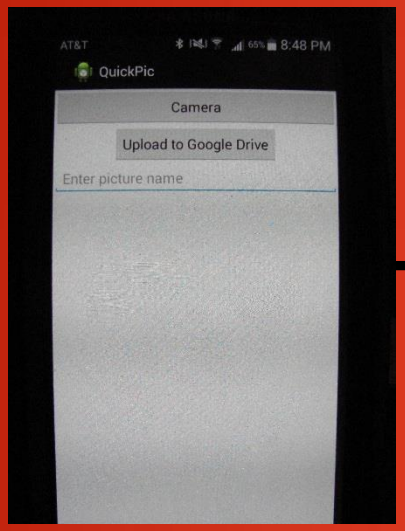
- Take a picture with your Android phone's camera
- Immediately upload to a private directory

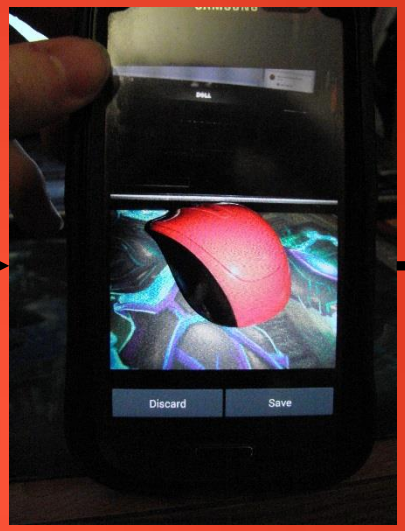
This project idea came up during one of our first classes, when we were brainstorming ideas and wanted to save our list digitally somewhere we could all access.

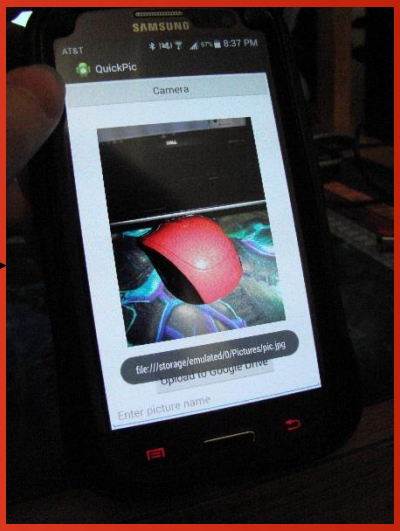
Project Features

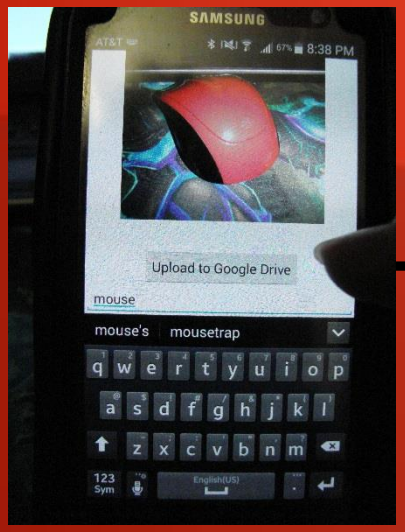
- Full access to Android phone’s camera, including light settings, filters, etc.
- Name your file and upload to Google Drive

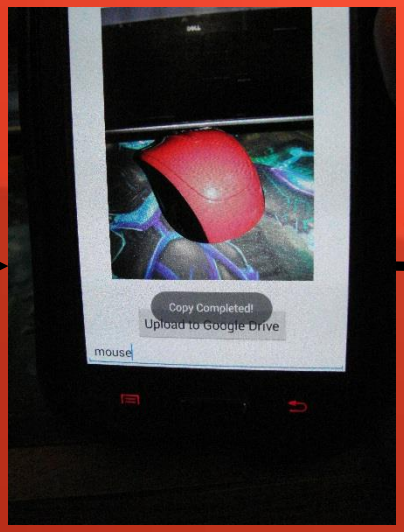
Usage

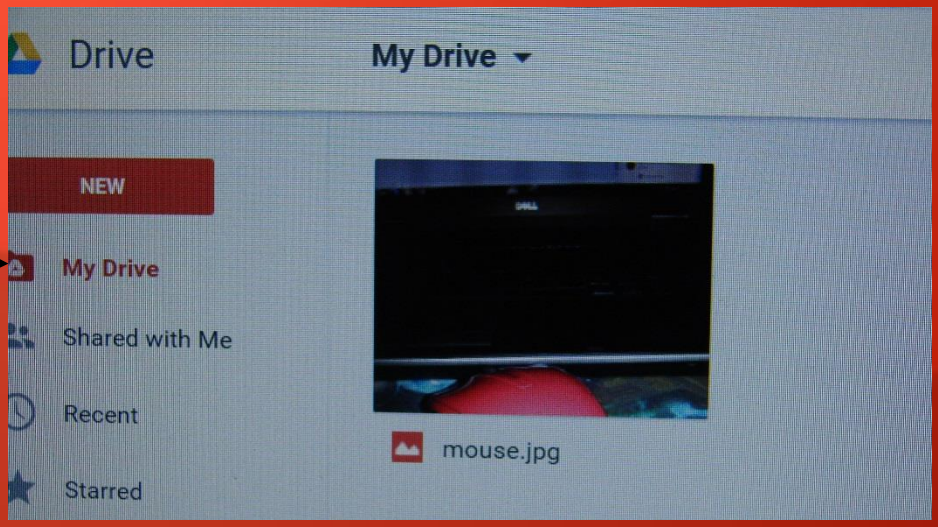

AT&T
8:48 PM
QuickPic
Camera
Upload to Google Drive
Enter picture name


Discard
Save


SAMSUNG
AT&T
8:37 PM
QuickPic
Camera
file:///storage/emulated/0/Pictures/pic.jpg
Enter picture name


SAMSUNG
AT&T
8:38 PM
Upload to Google Drive
mouse
mouse's
mousetrap


Copy Completed!
Upload to Google Drive
mouse


Drive
My Drive
NEW
My Drive
Shared with Me
Recent
Starred
mouse.jpg

Design and Implementation Details

Camera Interface

- We link directly into the Android camera, allowing access to all of its features. After a picture is taken, it is displayed in an imageView which allows the option to either discard or save it. Saving the image places it in a default location on the phone's memory card, from where it can be accessed and uploaded upon returning to the main screen.

Design and Implementation Details

Shared Preferences

- All account information that can be is stored internally. The user would be asked to enter this once, the first time, and then it could be automatically read and filled in every time after.

The OAuth process

- Google provides a reusable refresh token in addition to the one-time only access token. So after the first time performing OAuth, this too can be stored. The entire Google upload can then be done automatically.

Development Log

Week 5

- Camera Interfacing – Harshil

  -We had code to connect to the Android's camera almost immediately after starting the project. After taking a picture and choosing to save it, it is both stored to the phone's memory card and displayed on an ImageView

Week 7

- File uploads from the Eclipse console – Zak

  -We started coding on the console, before needing to worry about Android. The original plan was to upload to both Google Drive and Dropbox, so we have Temboo code that does both. This involved rewriting the POST routine and finding a better way to do the Base64 Encoding than we used in previous assignments.

Development Log

Week 9

- Webview code – Sam

  -When we discovered that Dropbox didn't provide a refresh token like Google, we started to look at Webviews in order to incorporate the internet and signing in to do OAuth every time. Due to technology and time constraints, we were unable to fully test and incorporate this, and the Dropbox functionality had to be dropped.

Week 10

- Transfer to Android – Zak and Harshil

  -We were weakened because we don't all have Android phones, and faced issues with driver incompatibility and memory allocation when we tried to connect with what we did have.

  -At first, we were getting the dreaded Dalvik error, from having multiple imports of the same jar files. If you import external jars, then rebuild the project any time after, they will get double-included and must be removed to get the code to compile.

Development Log

Week 10

- Transfer to Android – Zak and Dana

  -The (Almost) Project-Ending Error: “Could not find method xxx, referenced from method yyy”. We encountered this twice, with our Base64 Encoder and with JSONObjects. Android automatically includes a list of classes with every build it does, and unfortunately, most of them have names which conflict with other common imports. At compile/run time, the classloader will randomly pick which version of the conflicting class to use…and it usually picks wrong. This leads to it not being able to find and run methods of the imported class because they don’t exist in the Android default class. We fixed the Base64 error by changing the method we called from our import to an equivalent one which existed in Android’s imports.

Development Log

Week 10

- Transfer to Android – Zak and Dana

  -However, we could not do the same for the JSONObject error, because *nowhere in our code do we explicitly mention JSONObjects*. The conversion it is trying to do occurs within a Temboo choreo, and not on an accessible level.

  -There is no way to force the classloader to pick our import over the default or exclude classes from the default, but the solution turned out to be much simpler: We needed a different set of jar files for Android development than we did in Eclipse. Using the correct jars solved all of our problems.

  -This problem seems trivial, but it took us more time than any other aspect of the project to solve, and caused us to abort the Dropbox webviews and additional planned functionality.

- Aesthetic Design – Zak and Dana

  -The last step was to clean up the layout, resize and orient the imageView, and allow renaming of the picture before upload.

Demonstration and Closing Remarks

The final app is on a scale much reduced from our initial design, but the included functionality both works well and is in the spirit of the original idea. Our team is proud to have produced the QuickPic app.

References and Timeline

Camera Examples:
http://www.tutorialspoint.com/android/android_camera.htm
http://stackoverflow.com/questions/2729267/android-camera-intent/2737770#2737770
Android Default Class Index:
http://developer.android.com/reference/classes.html
Solving the Base64 Error:
http://stackoverflow.com/questions/2047706/apache-commons-codec-with-android-could-not-find-method

Our Source Code is Located at:
https://bitbucket.org/Zak_Olyarnik/zwo24-cs275-winter2015/src/9649f304d180d9de5433d6e468b3053fa6875900/P3/QuickPic/?at=master

Camera Interface	~ 3 hrs
Console Code	~ 2 hrs
Setting Up Phone/Eclipse Interaction	~ 6 hrs
General Transfer to Android	~ 2 hrs
Trying to Solve Import Problems	~15 hrs
Documentation	~ 4 hrs
Cumulative	~ 32 hrs