UNIVERSITY OF SOUTHERN INDIANA

WE'RE MOVING TO AN AGILE METHODOLOGY FOR SOFTWARE DEVELOPMENT.
I DON'T KNOW ALL OF THE DETAILS, BUT I THINK ONE OF YOU HAS TO BE DESIGNATED THE SCRUMBAG.
DOES THAT SOUND RIGHT?
IT'S BETTER THAN I EXPECTED.

Dilbert.com @ScottAdamsSays

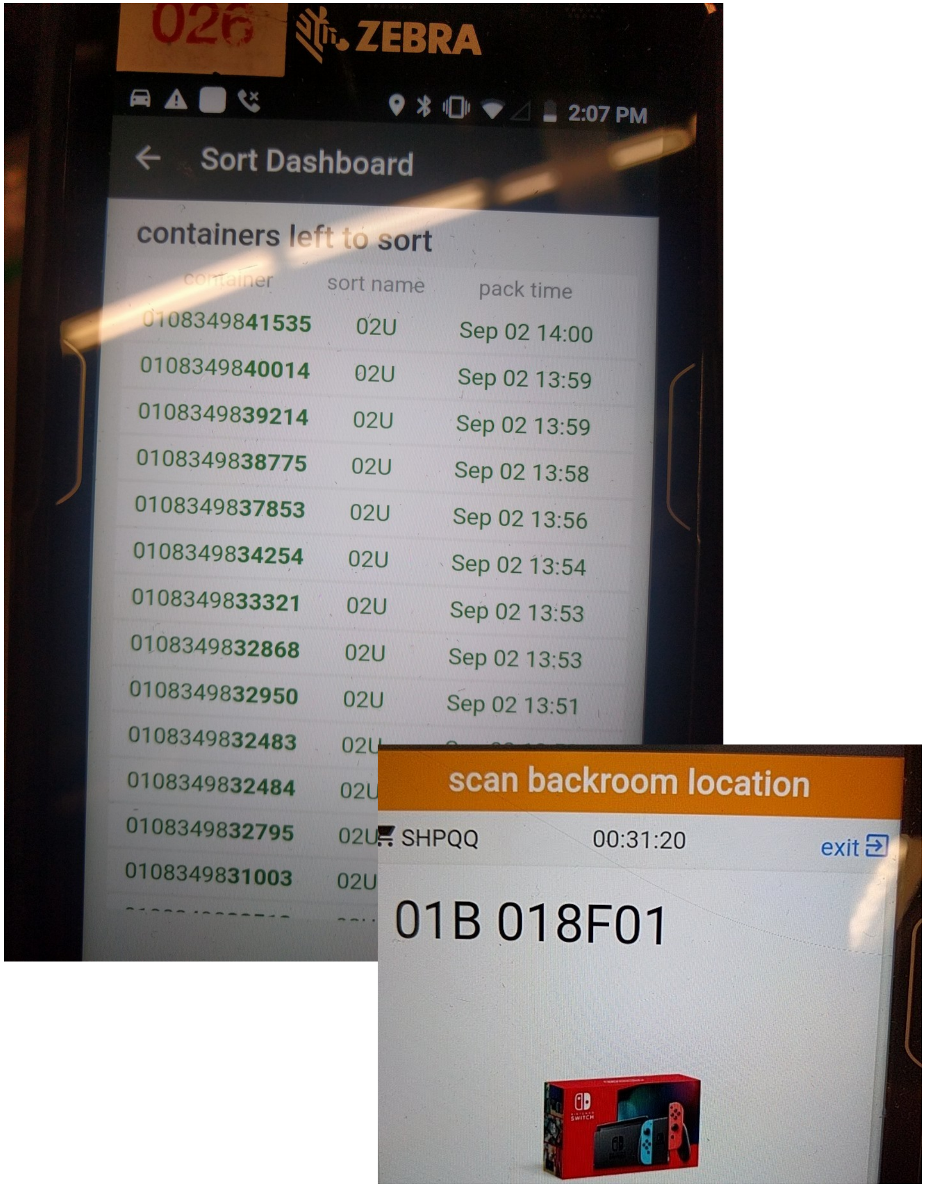
2-6-17 © 2017 Scott Adams, Inc./Dist. by Andrews McMeel

CS478 Software Development

Rockford Hipp
Sean Dodson
Ryan Bailey
Duane Bean

Project Bullsort (B.S.)

Sort Screen on Work Device



Sort Dashboard

containers left to sort

container	sort name	pack time
0108349841535	02U	Sep 02 14:00
0108349840014	02U	Sep 02 13:59
0108349839214	02U	Sep 02 13:59
0108349838775	02U	Sep 02 13:58
0108349837853	02U	Sep 02 13:56
0108349834254	02U	Sep 02 13:54
0108349833321	02U	Sep 02 13:53
0108349832868	02U	Sep 02 13:53
0108349832950	02U	Sep 02 13:51
0108349832483	02U	
0108349832484	02U	
0108349832795	02U	
0108349831003	02U	

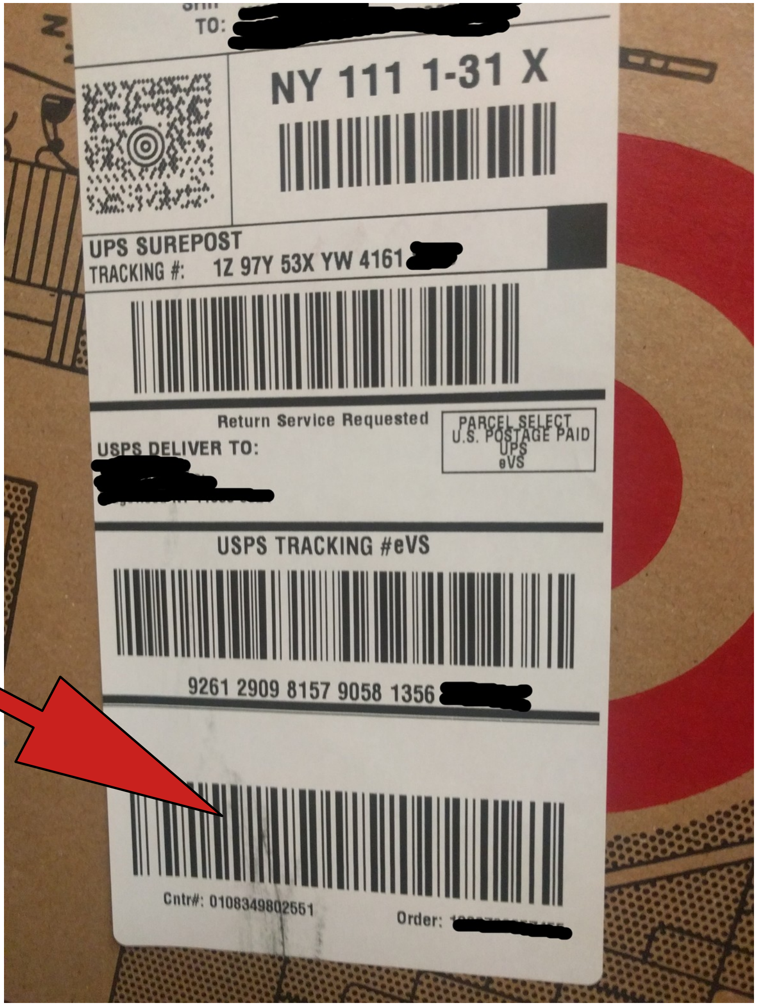
scan backroom location

SHPQQ 00:31:20 exit

01B 018F01

- The goal of our application is to capture images of all containter #s to sort
- Generate all of the barcodes captured
- Can also be used for backroom locations and item numbers.

Container #s on Label



SHIP TO: [redacted]

NY 111 1-31 X

UPS SUREPOST
TRACKING #: 1Z 97Y 53X YW 4161[redacted]

Return Service Requested

USPS DELIVER TO:
[redacted]

PARCEL SELECT
U.S. POSTAGE PAID
UPS
eVS

USPS TRACKING #eVS

9261 2909 8157 9058 1356[redacted]

Cntr#: 0108349802551

Order: [redacted]

How should we start?

We know that our app will have these components:

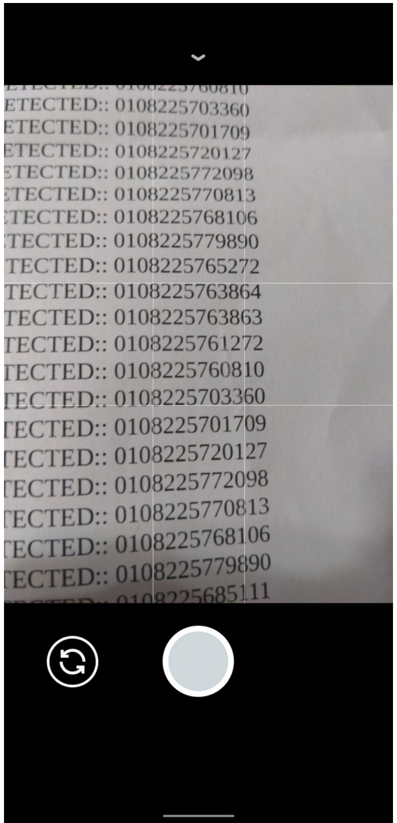
- Camera use
- Text recognition
- Barcode generator
- A view to display the barcodes & container#

To begin, RAD Screagles created a Slack channel and began researching tutorials on camera functionality and requesting camera permissions.

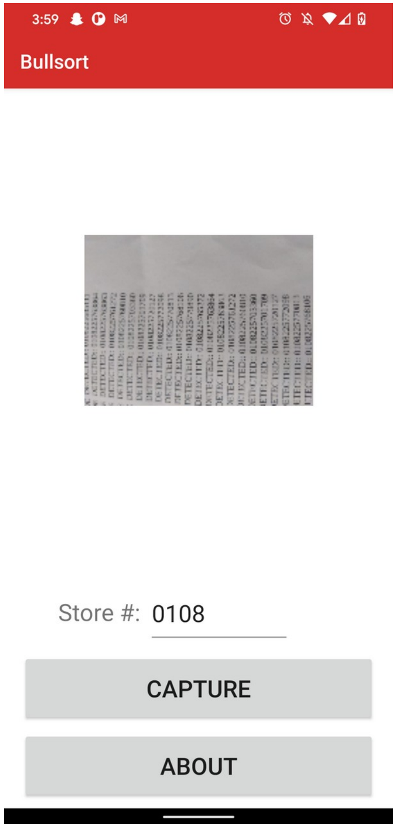
Basic Camera Test

- Android Studio Tutorial - Take picture with Camera learn android programming

  Although you’d have to manually enable the app’s camera permissions, we were able to get camera functionality by following the tutorial.



```
ETECTED:: 0108225703360
ETECTED:: 0108225701709
ETECTED:: 0108225720127
ETECTED:: 0108225772098
ETECTED:: 0108225770813
TECTED:: 0108225768106
TECTED:: 0108225779890
TECTED:: 0108225765272
TECTED:: 0108225763864
TECTED:: 0108225763863
TECTED:: 0108225761272
TECTED:: 0108225760810
TECTED:: 0108225703360
TECTED:: 0108225701709
TECTED:: 0108225720127
TECTED:: 0108225772098
TECTED:: 0108225770813
TECTED:: 0108225768106
TECTED:: 0108225779890
```

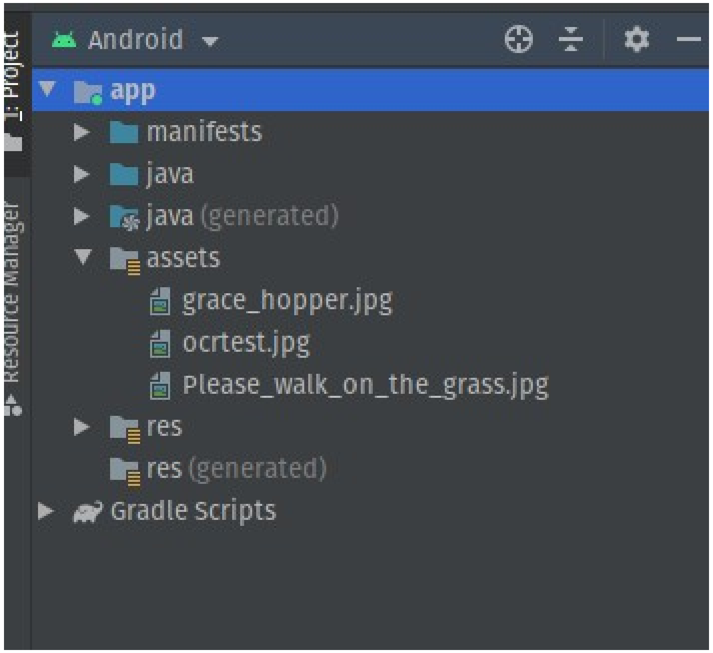


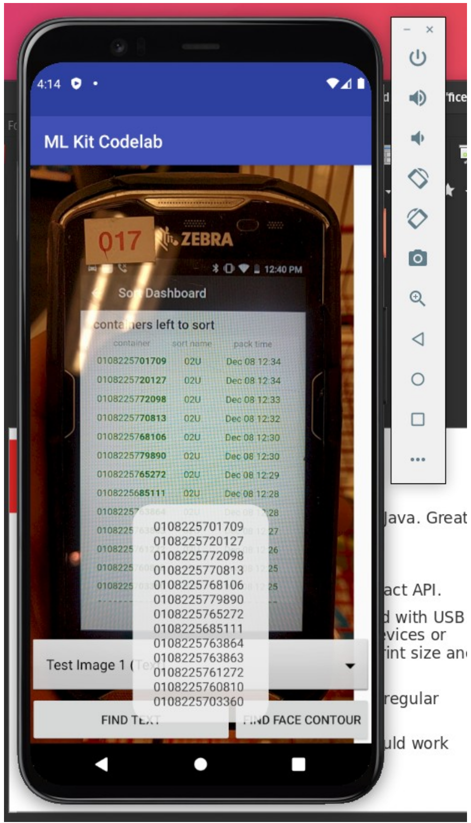
Store #: 0108

From Tesseract to ML Kit

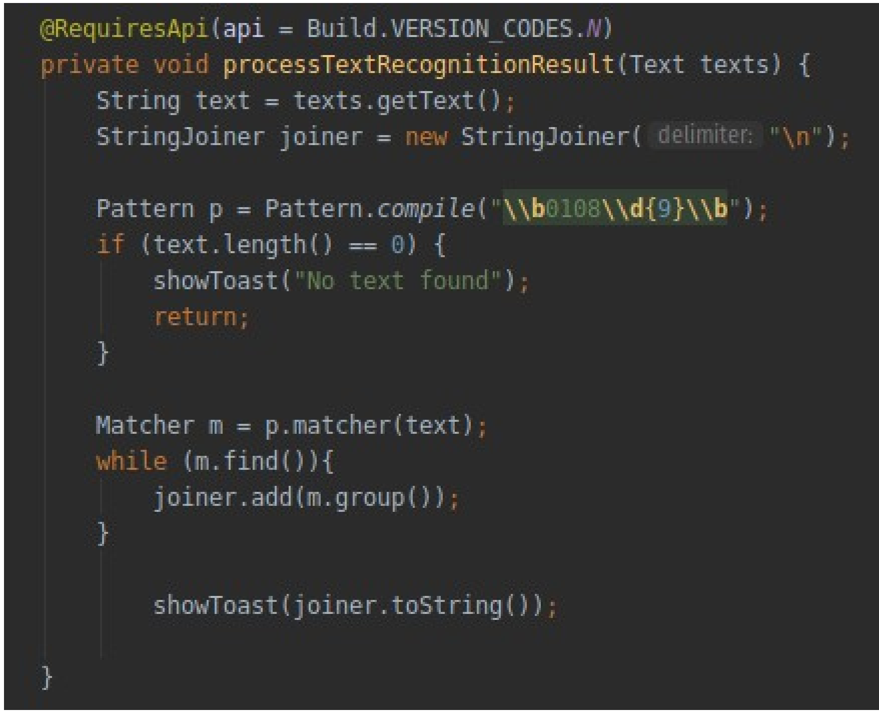
- https://codelabs.developers.google.com/codelabs/mlkit-android#4

  With a little research, the team discovered a codelabs tutorial on text recognition with ML Kit. By replacing the example picture with one inside assets folder, we can test the text recognition.



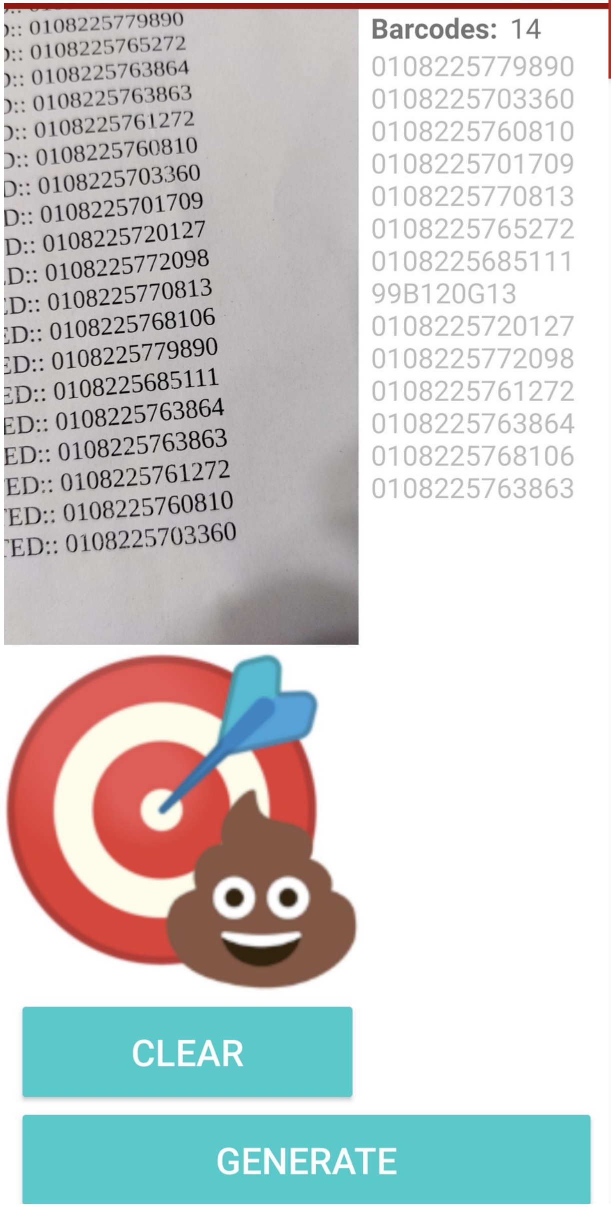


```
0108225701709
0108225720127
0108225772098
0108225770813
0108225768106
0108225779890
0108225765272
0108225685111
0108225763864
0108225763863
0108225761272
0108225760810
0108225703360
```



```
@RequiresApi(api = Build.VERSION_CODES.N)
private void processTextRecognitionResult(Text texts) {
    String text = texts.getText();
    StringJoiner joiner = new StringJoiner( delimiter: "\n");

    Pattern p = Pattern.compile("\\b0108\\d{9}\\b");
    if (text.length() == 0) {
        showToast("No text found");
        return;
    }

    Matcher m = p.matcher(text);
    while (m.find()){
        joiner.add(m.group());
    }

        showToast(joiner.toString());

}
```

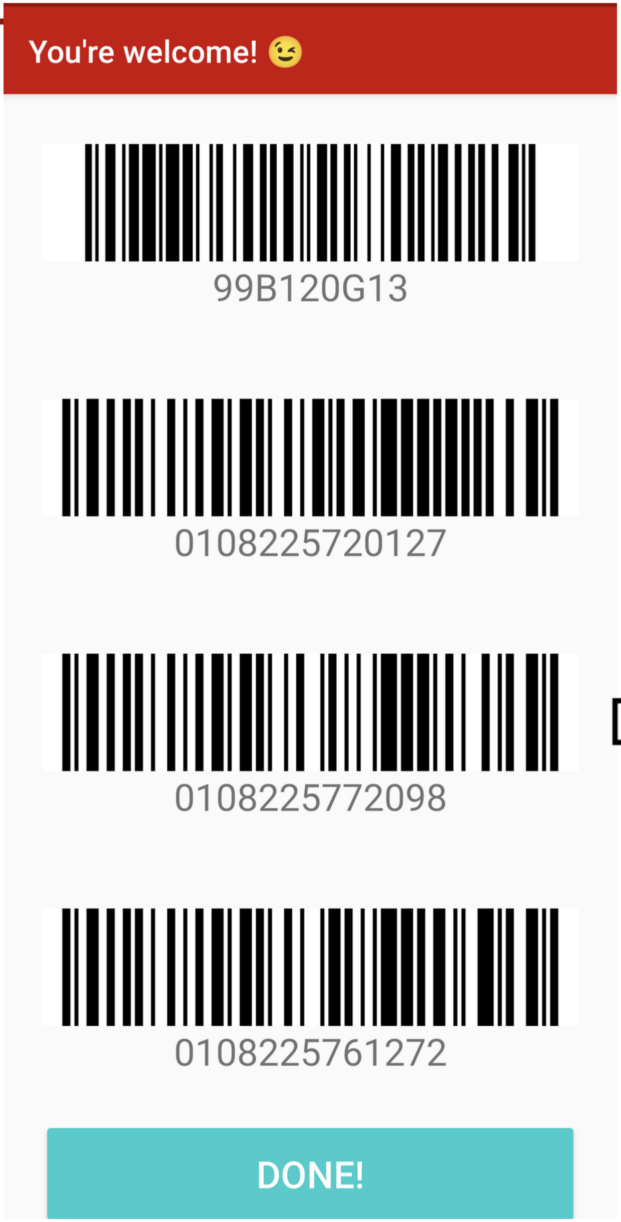
App Design



):: 0108225779890
):: 0108225765272
):: 0108225763864
):: 0108225763863
):: 0108225761272
):: 0108225760810
D:: 0108225703360
D:: 0108225701709
D:: 0108225720127
D:: 0108225772098
D:: 0108225770813
ED:: 0108225768106
ED:: 0108225779890
ED:: 0108225685111
ED:: 0108225763864
ED:: 0108225763863
ED:: 0108225761272
ED:: 0108225760810
ED:: 0108225703360

Barcodes: 14

0108225779890
0108225703360
0108225760810
0108225701709
0108225770813
0108225765272
0108225685111
99B120G13
0108225720127
0108225772098
0108225761272
0108225763864
0108225768106
0108225763863



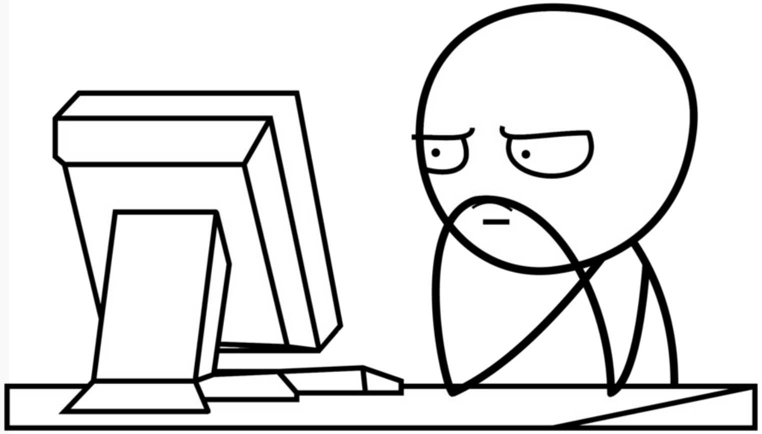
You're welcome! 😉

99B120G13

0108225720127

0108225772098

0108225761272



From Java to Kotlin

- At this point our team decided to switch from Java to Kotlin.
- CameraX is important because it tries to simplify app dev for all devices and system builds. Getting Started with CameraX demonstrates

1. Requesting permissions
2. Camera preview, image analysis, & capture
3. Kotlin language

- Any Java code copied into the IDE is reformatted for Kotlin

Next Steps

- How to determine the application’s data directory?
- Generate the barcodes
- Ensure they are saved to the app’s data directory (Cannot use ADB shell if not requesting external storage permission)
- Place barcodes in a RecyclerView on a new activity screen.

  Android RecyclerView Tutorial with Kotlin

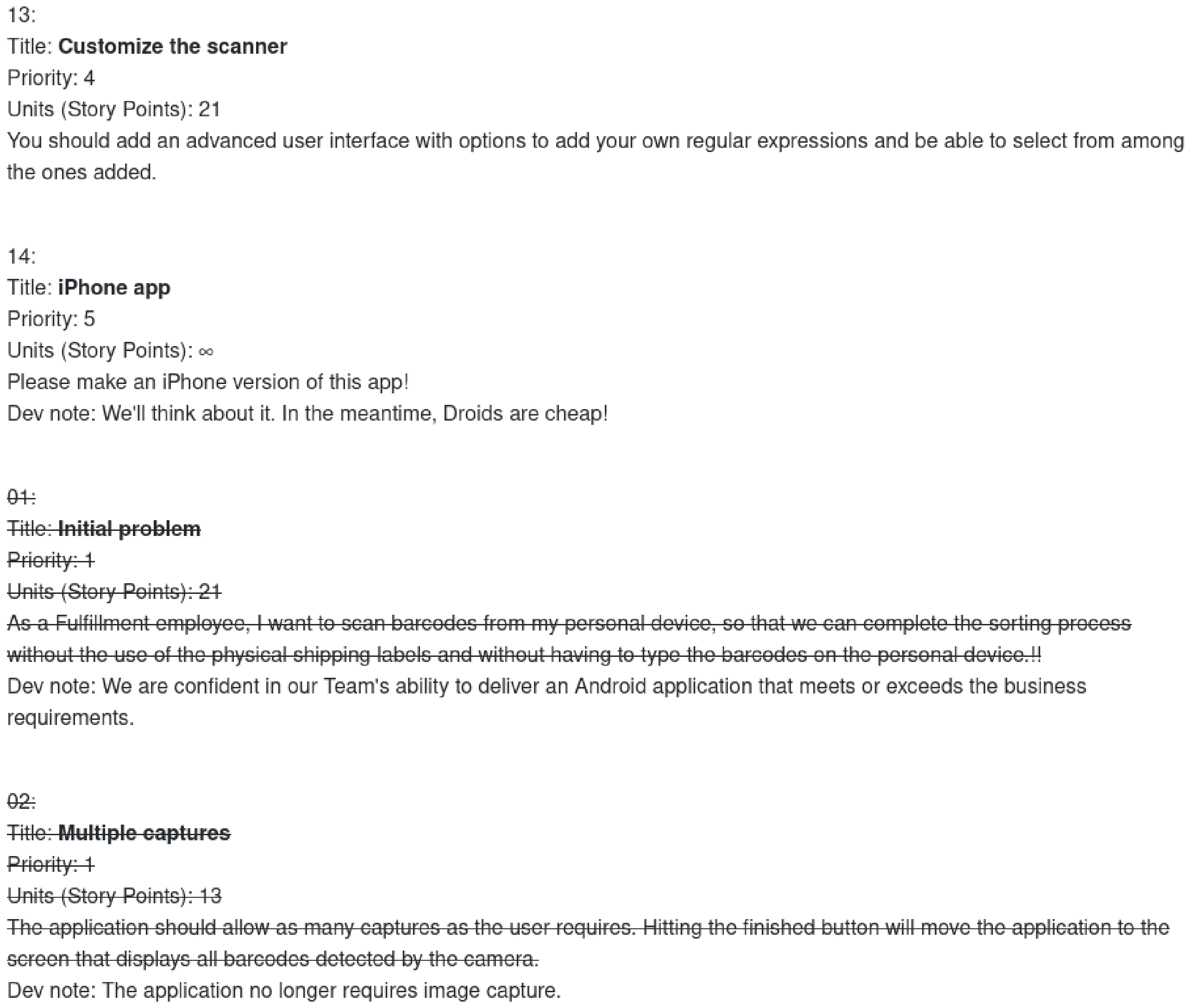
Next Steps

- Camera analysis runs on a thread separate from the main, so the Team will need to update the list of barcodes found and the count from the main.

```
if (concurrentHashSet.size > 0) {
    val newText = concurrentHashSet.joinToString(
        prefix = "",
        postfix = "",
        separator = "\n"
    )
    txtBarcodes.setText(newText)
    txtCount.text = concurrentHashSet.size.toString()
}
```

Fav Github Features

- Story revision during app progression



13:
Title: **Customize the scanner**
Priority: 4
Units (Story Points): 21
You should add an advanced user interface with options to add your own regular expressions and be able to select from among the ones added.

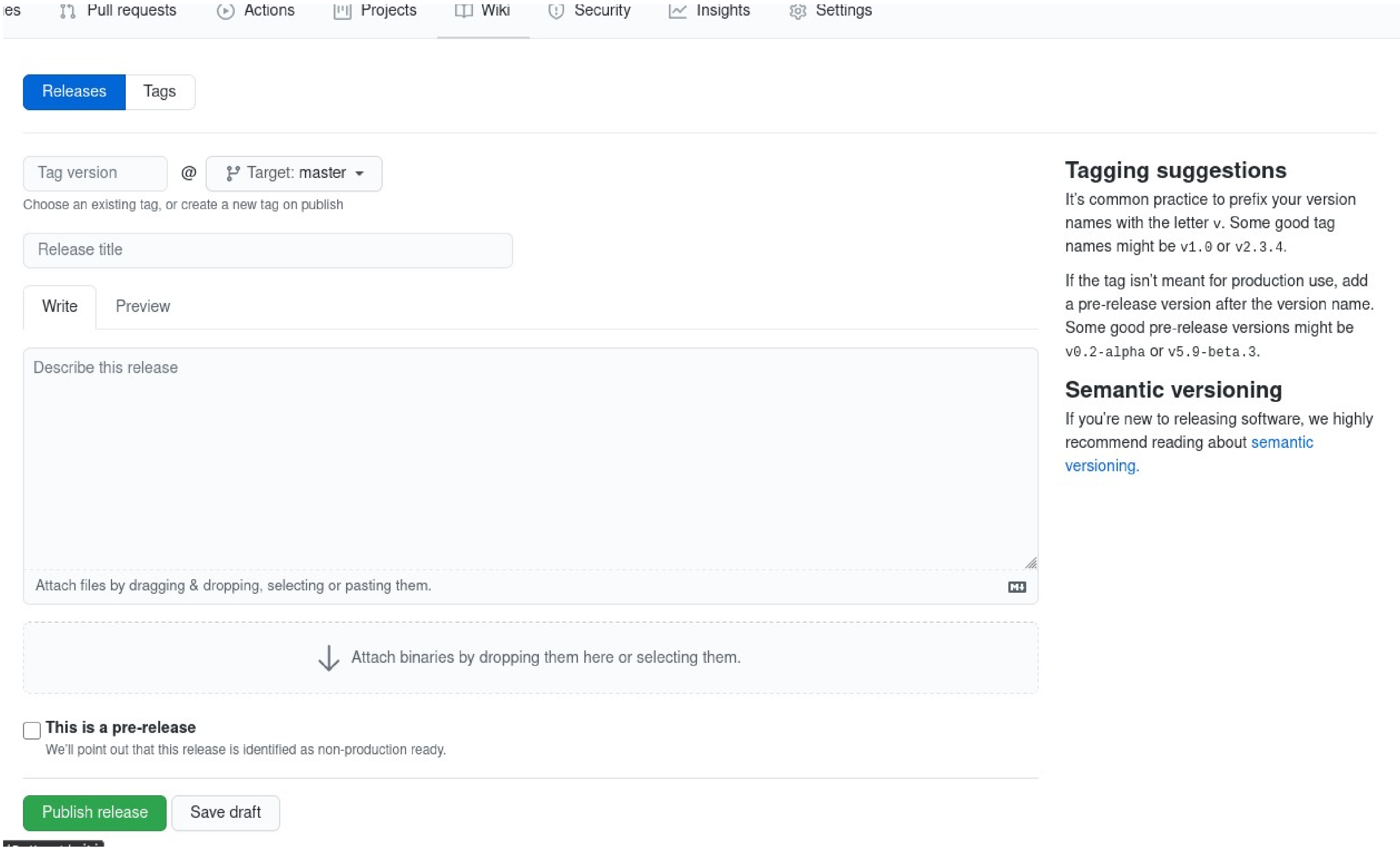
14:
Title: **iPhone app**
Priority: 5
Units (Story Points): ∞
Please make an iPhone version of this app!
Dev note: We'll think about it. In the meantime, Droids are cheap!

~~01:~~
~~Title: **Initial problem**~~
~~Priority: 1~~
~~Units (Story Points): 21~~
~~As a Fulfillment employee, I want to scan barcodes from my personal device, so that we can complete the sorting process without the use of the physical shipping labels and without having to type the barcodes on the personal device.!!~~
Dev note: We are confident in our Team's ability to deliver an Android application that meets or exceeds the business requirements.

~~02:~~
~~Title: **Multiple captures**~~
~~Priority: 1~~
~~Units (Story Points): 13~~
~~The application should allow as many captures as the user requires. Hitting the finished button will move the application to the screen that displays all barcodes detected by the camera.~~
Dev note: The application no longer requires image capture.

Fav Github Features

- Creating releases



Choose an existing tag, or create a new tag on publish

Attach files by dragging & dropping, selecting or pasting them.

Attach binaries by dropping them here or selecting them.

□ **This is a pre-release**

We'll point out that this release is identified as non-production ready.

Tagging suggestions

It's common practice to prefix your version names with the letter `v`. Some good tag names might be `v1.0` or `v2.3.4`.

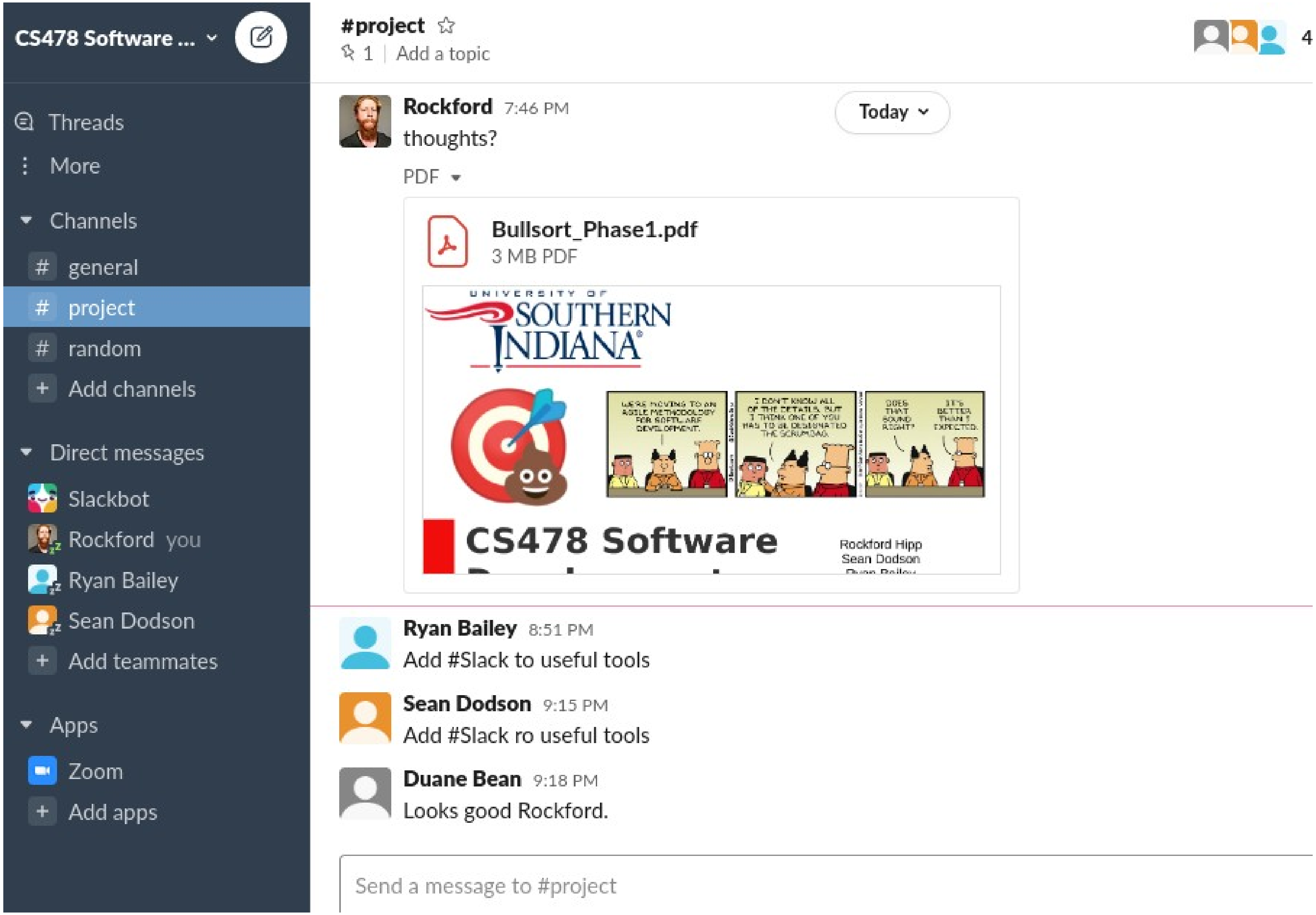
If the tag isn't meant for production use, add a pre-release version after the version name. Some good pre-release versions might be `v0.2-alpha` or `v5.9-beta.3`.

Semantic versioning

If you're new to releasing software, we highly recommend reading about semantic versioning.

Useful Tools

- #Slack!



Rockford 7:46 PM
thoughts?

Bullsort_Phase1.pdf
3 MB PDF

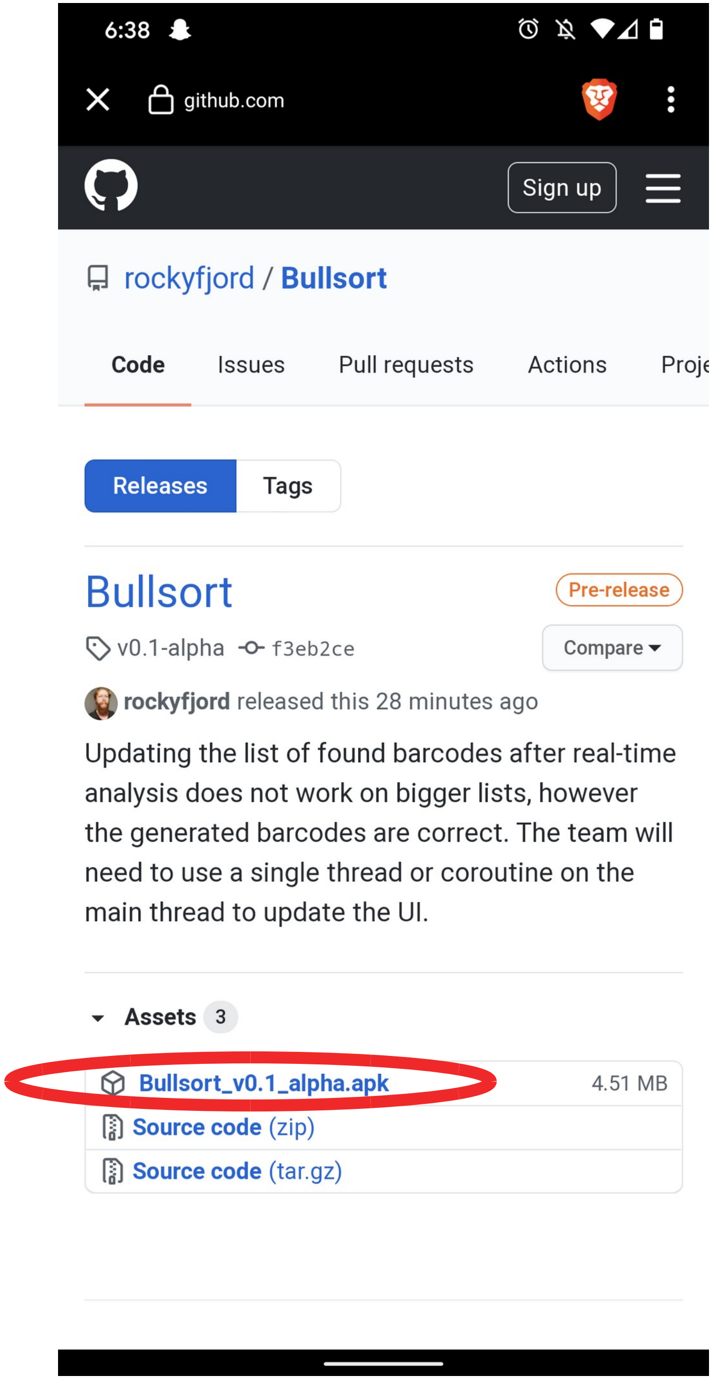
Ryan Bailey 8:51 PM
Add #Slack to useful tools

Sean Dodson 9:15 PM
Add #Slack ro useful tools

Duane Bean 9:18 PM
Looks good Rockford.

Send a message to #project

Try the App!



rockyfjord / **Bullsort**

Bullsort

v0.1-alpha f3eb2ce

rockyfjord released this 28 minutes ago

Updating the list of found barcodes after real-time analysis does not work on bigger lists, however the generated barcodes are correct. The team will need to use a single thread or coroutine on the main thread to update the UI.

Assets 3

Bullsort_v0.1_alpha.apk	4.51 MB
Source code (zip)	
Source code (tar.gz)	