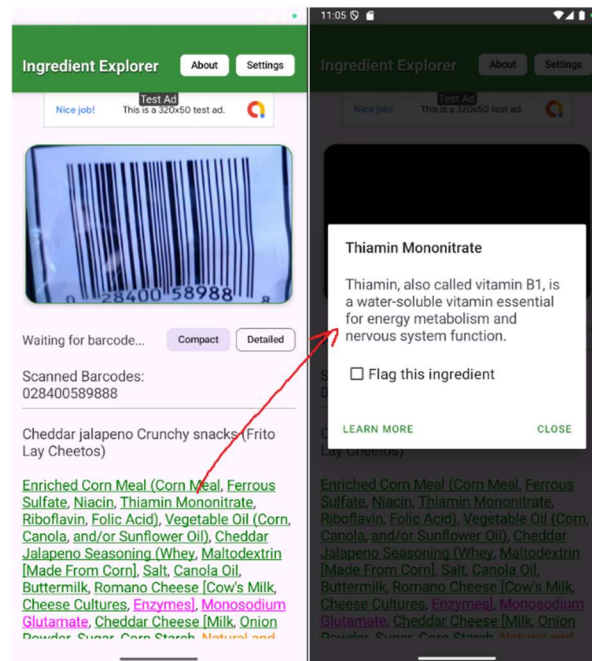


Artifact Two Narrative - OCR

My artifact for this enhancement is a new OCR (Optical Character Recognition) class for my android app Ingredient Explorer. I began working on the app a couple months ago based on an app UI framework I made in CS-319 (UI/UX Design/Development) and Android Studio experience in CS-360 (Mobile Architecture). It's designed to scan food product barcodes, retrieve their ingredient list, then users can touch an ingredient to learn about it. It gets the products ingredient list by looking up the scanned UPC codes which requires internet access, so my enhancement is letting users input ingredients lists into the app by taking a picture of it.

Typical app usage:



I chose this artifact because it shows my ability to manage complex data, make it compatible with an existing system, and integrate it as a new function in the system. Text initially retrieved from OCR usually includes random extraneous characters and spelling errors like replacing B with 8, E with 3, or O with 0, so cleaning it algorithmically takes some

consideration. This artifact meets my planned course outcomes by designing, evaluating, and implementing algorithms that process unstructured real-world data into a usable, structured form.

Raw OCR scan results in-app:



While working on this enhancement I had to test a few different approaches before reaching a good design. After initially getting the camera feed with OCR working it came up with a lot of undesired text, so my first idea was to let the user crop the image to the ingredient list, but the scope of that got excessive needing new UI tools and image manipulation and it ended up not really being a good idea from a user experience perspective. I arrived at the solution of ignoring OCR'd text until the word "ingredient" was found, since that's universally at the start of ingredient lists. The function reads text after the ingredient list starts, and checks for words which commonly follow it to stop reading the text. Another issue I came across was the problem with OCR swapping letters and numbers, initially importing a spell-checking library

seemed like a good idea but for this application it ended up being too slow and resource intensive. I ended up simply replacing numbers with letters in the scanned text, since numbers are rarely found in product ingredient lists it's a simple but effective solution.

