Task: Reverse engineering a problem for a solution

Verifying documents is a huge area known as "KYC" know your customer, and "IDV" identity verification. You'll do this when you are signing up online for a bank account etc.

For this task you'll need a passport and an iPhone.

Please spend no more than 4 hours in total (up to 3 hours reading, reviewing and modifying code to determine how things might work, and 1 hour reflecting on how you might do the rest).

To get an idea of what we would like to do, install the "Innovatrics DOT" app from the app store. It features an onboarding process which scans the MRZ zone from your passport, then reads the NFC chip.

Install the "Innovatrics DOT" app
https://apps.apple.com/us/app/innovatrics-dot/id1495147772
- Choose full onboarding and scan the front side of your passport
- Try different angles and lighting conditions, and note the feedback given to the user
- Cancel the second side scan and note the MRZ machine readable zone data extraction

Goal
Work out the steps required to create an MRZ scanner like the DOT MRZ reader - only the part that detects the MRZ text rectangles, and an additional part that gives feedback to the user when it can't read.

We are interested in what you understand about the problems that need to be resolved to create such a solution, what problems you encounter, other problems you don't encounter but consider, and what range of solutions you might think up to resolve them.

For instance, you can just put marker comments in the code like "to give feedback to the user on the document not being centred - a calculation that the bounding boxes are too far left/too far right would go here".

Watch this talk to get an overview of the process (this outlines how the live camera stream and recognition work); but here "machine readable", refers to barcodes, not MRZ text.

Capture machine-readable codes and text with VisionKit - WWDC22 - Videos
https://developer.apple.com/videos/play/wwdc2022/10025/

Download Apple's vision text recognition sample app code here:
Extracting phone numbers from text in images | Apple Developer Documentation
https://developer.apple.com/documentation/vision/extracting-phone-numbers-from-text-in-images

1. Modify the sample code to identify MRZ zone text instead of phone numbers. Unless your passport has umlauts in your name, a simple "strip spaces, look for a "<" in (the first line) of length 44, 36, or 30 characters; and then 2 (passport) or 3 remaining lines of the same length (44 for a passport).

   Confirm you can dump your MRZ lines to the console, or show them in the overlay.

2. Implement a better overlay region for an ID card / passport with feedback. Show the feedback "success" or the passport number, the first time the MRZ lines are correctly detected so we can see how quickly this occurs without fully parsing the MRZ, or watching the logs.

Consider the following points for discussion
(Code is not expected unless you have time / want to)

- How could you give feedback to the user on whether the document is too far away or too close? (Hint, we didn't use text rectangles for this, but you could.)

- How could you give feedback to the user on whether the image has too much reflection, is too bright or too dark?

- How could you test the above scenarios (document too far, too close, too bright) to check for changes within the app?

- How could you test the above scenarios (document too far, too close, too bright) to compare our implementation, and detection speed, to the Innovatrics DOT app?

- How can we test the apps response bad / partial / malformed MRZ text lines?

- What optimizations could be made to improve the text detection speed?

- What are the overall steps of the process - (this more identifies what parts of the business logic might be identical, or with KMP shared) between Android / iOS?

- MRZ is a standard, but has variations - eg. Russian Identity cards, and character transliteration. How might we handle a situation where a customer comes to us and says "passports aren't scanning and our passengers are getting frustrated"?

- When deciding whether to buy an off the shelf solution like Innovatrics DOT, or create our own, what would you consider after understanding more about the effort and maintenance costs?

Bonus points, security

- The second stage of passport / identity verification is using the MRZ data to unlock the NFC chip. This validates that a person is in physical possession of that passport.

If we can match their name on a booking, to the name read from the passport, how sure can we be (what commitment can we make as vendor, to our customers) that the identity document isn't fraudulent? So - this is a typical "put yourself in the mindset of a criminal" - what ways can you think of that you could/would you defraud this process knowing the steps involved?

- If we are asked to store the data, or a subset of the data read from the passport, so it can be re-used without going through the process again, what precautions could or should be taken?

More resources
ICAO 9303 mrz standard https://en.wikipedia.org/wiki/Machine-readable_passport
An MRZ parser written in Java by the Innovatrics DOT people from above is https://github.com/ZsBT/mrz-java
PRADO Public Register of Authentic Documents Online
https://www.consilium.europa.eu/prado/EN/prado-start-page.htm

Share code
Upload your code modified from the Apple sample to a github repository so that we can compile it, and share or send the link prior to the follow up meeting if possible.