This document explains the problem to solve, the data sources available, and the strategy to solve the problem.

Problem to solve

The airports these days have two main problems: reduce queue times and keep the distances between people because of the pandemic issue that is afflicting the world. An automated kiosk that automates the check-in process of the passengers is a great tool to solve those problems, and because it's automated it's also less error-prone compared to the manual activities of the operators. This last aspect makes it a good choice even in terms of security.

The Automated Kiosk should work as follows:

- Read all the relevant data in the Boarding Pass and the ID Card of the passenger using a camera
- Compare and validate the data of the Boarding Pass with the ID Card
- Compare and validate this data with the data in the internal systems of the airport to make sure the passenger is supposed to be onboarded
- Verify the passenger identity by comparing the ID Card photo with the passenger face captured by recording a 30-seconds video
- Verify the luggage doesn't contain any lighter
- Display a message to the user to inform about the result of the check-in process

Data Sources

- 5: ID Cards of the passengers
- 5: Boarding Passes of the passengers
- 1: 30-seconds video of one passenger
- 45 Lighters images for training and 5 for test purpose

Solution strategy

The Automated Kiosk takes advantage of a combination of Azure Cognitive Services linked together using Python code. The Azure Cognitive Services to use are:

- Azure Blob Storage: This is required to store the training dataset and the manifest file
- Azure Form Recognizer: This is required to recognize and get the data from the Boarding Pass and ID Card. For the ID Card we can use a pre-built model and feed it with the 5 ID Cards to train it. The model to extract data from the Boarding Passes is Custom, and it's created using the Boarding Passes available in the Azure Blob Storage.
- Azure Video Analyzer: This is required to analyze the 30-seconds video of the passenger to get thumbnails, the emotions, the sentiments, and other data extracted from the video
- Azure Face API: This is required to get the face of the passenger from the ID Card and the video and finally compare. It takes the ID Card image and compares it with the image extracted from the 30-seconds video.
- Azure Custom Vision: This is required to check if there are lighters in the luggage. The model is trained using the 45 lighters images and tested using the 5 test images. The trained model has an accuracy that is greater than 75%.

The Python scripts allow to:

- Get the data from the ID Card and Boarding Pass using the Azure Form Recognizer
- Compare this data to make sure they are consistent. If they are not consistent, the passenger is invited to go to a customer service representative.
- Compare this data with the data available in a CSV file hosted in the internal systems of the Airport. If not found in the manifest file, the passenger is invited to go to a customer service representative.
- Get face data from the 30-seconds video uploaded on the VideoIndexer.ai portal and connected with the Azure Video Analyzer resource
- Get the face from the ID Card using the Azure Face API and compare it with the face extracted from the 30-seconds video. For this application, they match if the confidence is greater than 0.65. If the confidence is lower, the passenger is invited to go to a customer service representative.
- Check if there are lighters in the photo of the luggage's content. This is done taking advantage of the Azure Custom Vision using a model trained on 45 lighters images. The returned value from the model on the Azure Custom Vision contains a probability. For this application, if the probability value of the presence of lighters in the image is greater that 0.65, we inform the passenger that the baggage has been flagged for removal.
- After all these checks, if everything is okay, a message with all the boarding data is displayed to the passenger.