RISK IDENTIFICATION FOR TEAM 06

1. Working with an OCR API – Adrian Lane
   1. Research the Azure cognitive services OCR API
   2. Create a small PoC to use as a reference for our app approach
2. Working with a natural speech engine API – Graham Walker
   1. Research the Azure cognitive services Text to Speech API
   2. Create a small PoC to use as a reference for our app approach
3. Cloud storage (database, blob, what?) – Ruben
   1. Dependent upon data model, but likely:
      1. Azure aBlob storage for stored images
      2. Azure Database for all other data persistence
4. Working with the camera/images API on the phone – Kenyon Bunker
   1. Research using the camera within the app - Done
   2. Research storing images from the camera elsewhere, (i.e. where our app would likely need to reference it from)
      1. I think internal storage should be fine
   3. Research accessing the photo album from within the app for our app’s use
      1. I think it should only be visible to the app
      2. And images to be removed if the app is uninstalled
   4. Create a small PoC to use as a reference for our app approach
   5. Update: <https://developer.android.com/guide/topics/media/camera>
   6. Update: Requirements
      1. Camera Requirement
         1. declare the [camera requirement in your manifest](https://developer.android.com/guide/topics/media/camera#manifest).
      2. Quick Picture
         1. [Using Existing Camera Apps](https://developer.android.com/guide/topics/media/camera#camera-apps).
      3. **Foreground Services Requirement – yes - completed**
      4. Storage
         1. Only visible to app
         2. Images removed if app is uninstalled.
         3. Images to be stored locally and in blob storage
         4. Images should have auto clean up
         5. One possibility for saving space is to adjust the photo to a smaller size before it is saved permanently. However, this requirement is dependent on what the OCR API will need
         6. See  [Saving Media Files](https://developer.android.com/guide/topics/media/camera#saving-media)