Team PYCO:
Edward Marecos, Akemi Sai

PYCO

Your Personal Fashion Companion

API

Google Cloud Vision API

detect and label the clothing item such as “jacket”, “shirt”, etc.

Primary benefits

- **Advanced Image Recognition**
  - The API can identify specific clothing time, colors, patterns within an image, which helps categorize and tag user-uploaded images effectively. This enhances the app’s search and organization features

- **Real-time Fashion Recommendations**
  - By analyzing the images of clothing, the Vision API can offer styling or outfit suggestions based on popular or similar items.

API

CameraX API

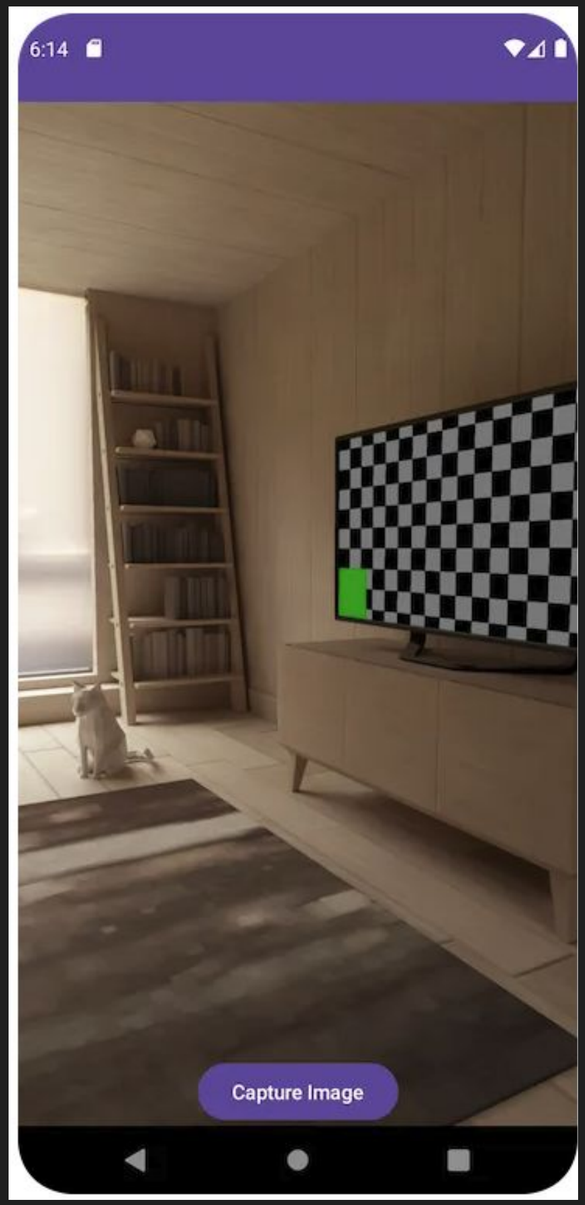
allows users to capture images of their clothing directly from the app

Primary benefits
CameraX improves the developer experience in several key ways.

Broad device compatibility
CameraX supports devices running Android 5.0 (API level 21) and higher, representing over 98% of existing Android devices.

Ease of Use
CameraX emphasizes use cases, which allow you to focus on the task you need to get done instead of managing device-specific nuances. Most common camera use cases are supported:

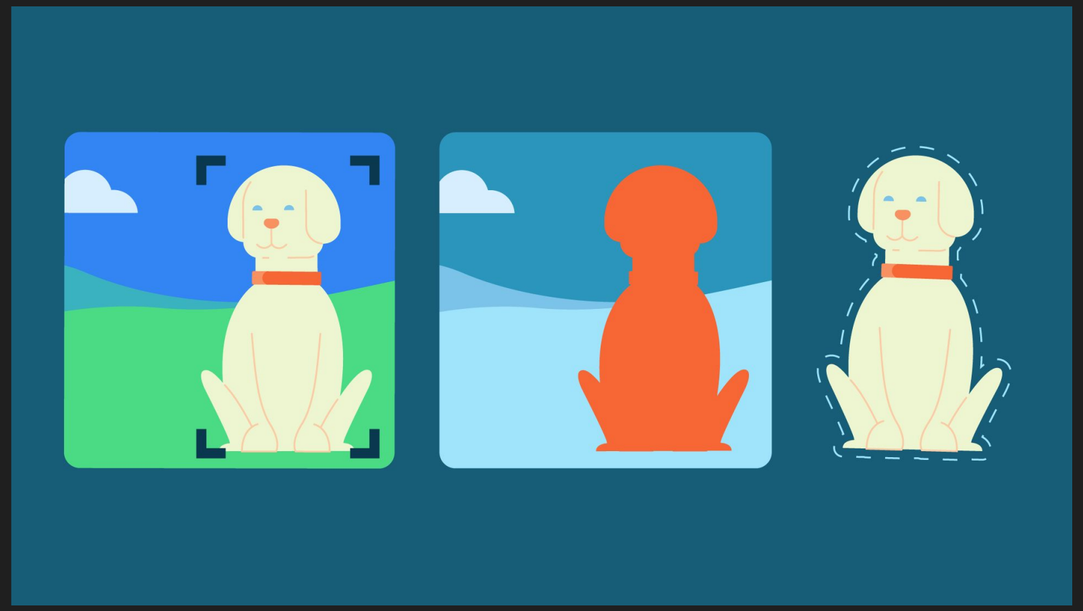
- Preview: View an image on the display.
- Image analysis: Access a buffer seamlessly for use in your algorithms, such as to pass to **ML Kit.**
- Image capture: Save images.
- Video capture: Save video and audio.



API

ML Kit

remove the background of the picture



Primary Benefits
ML Kit's subject segmentation API allows developers to easily separate multiple subjects from the background in a picture, enabling use cases such as sticker creation

Device Capability – Ease of Use
tailored to android, CameraX api is designed already to work with this.

Key capabilities
Multi-subject segmentation: provides masks and bitmaps for each individual subject, rather than a single mask and bitmap for all subjects combined.
Subject recognition: subjects recognized are objects, pets, and humans.
On-device processing: all processing is performed on the device, preserving user privacy and requiring no network connectivity.

Other APIs

WorkManager

monitors network status to trigger offline or online modes, using WorkManager to sync data as soon as a connection is restored

Firebase Firestore

used for storing user account information, clothing inventory, outfit requests and feed data.

On device sensors

Camera

to upload their images from their phone

Firebase Cloud Messaging

sends push notifications to users for events like new followers, likes, comments, milestones, etc.

Firebase Authentication

provides secure user authentication, ensuring that only authenticated users can access their accounts and features.

TikTok LoginKit

Firebase can fetch user details from Tiktok API

External Libraries

Retrofit

library for making HTTP requests and parsing responses

Moshi

library for JSON parsing to work with data from APIs

Room

local database library for offline data persistence

ML Kit

a mobile SDK library for on-device machine learning tasks, including image processing

Coil

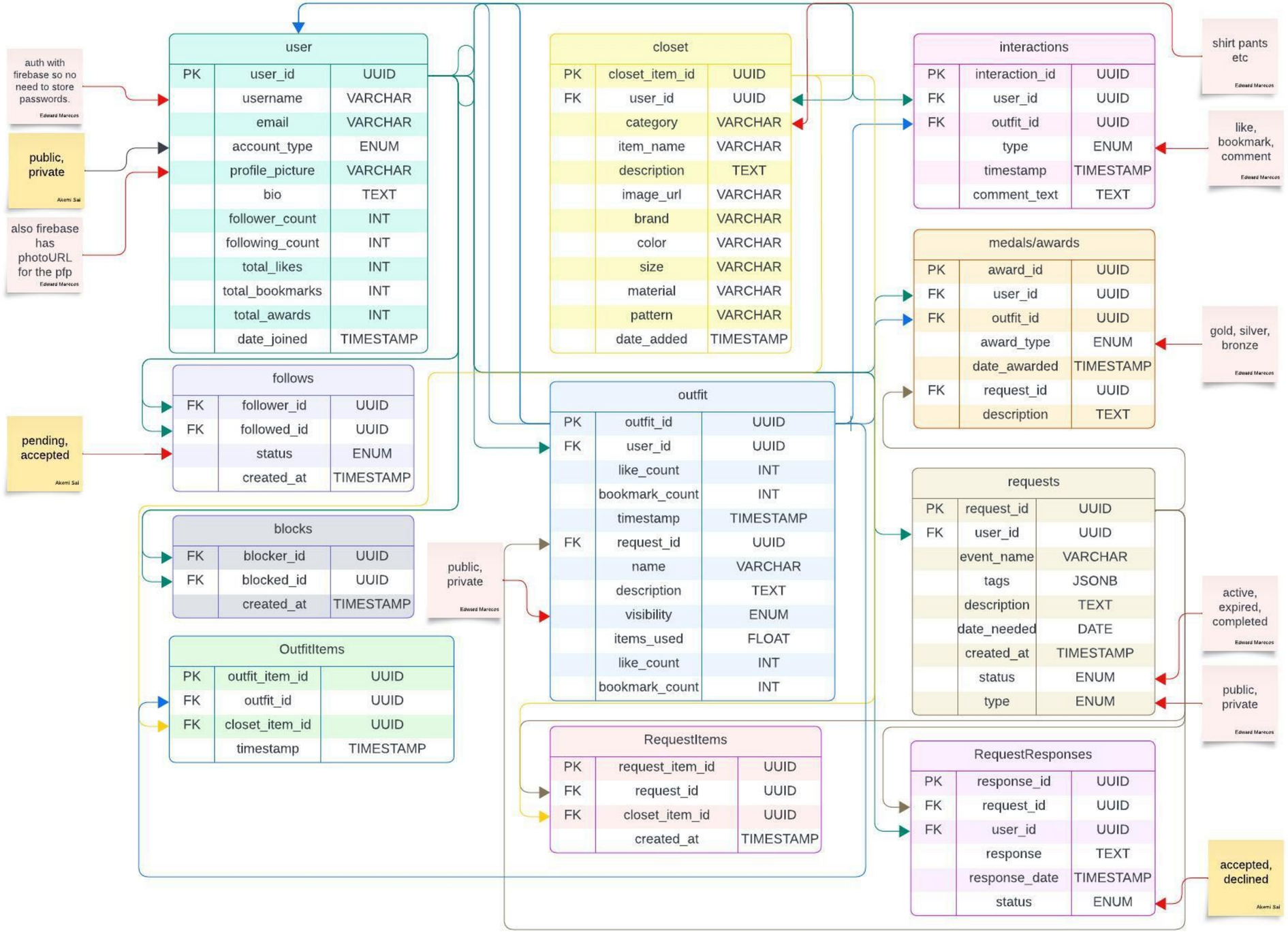
image loading and caching libraries.

WorkManager

a library for scheduling and managing background tasks

EncryptedSharedPreferences

library for securely storing sensitive data

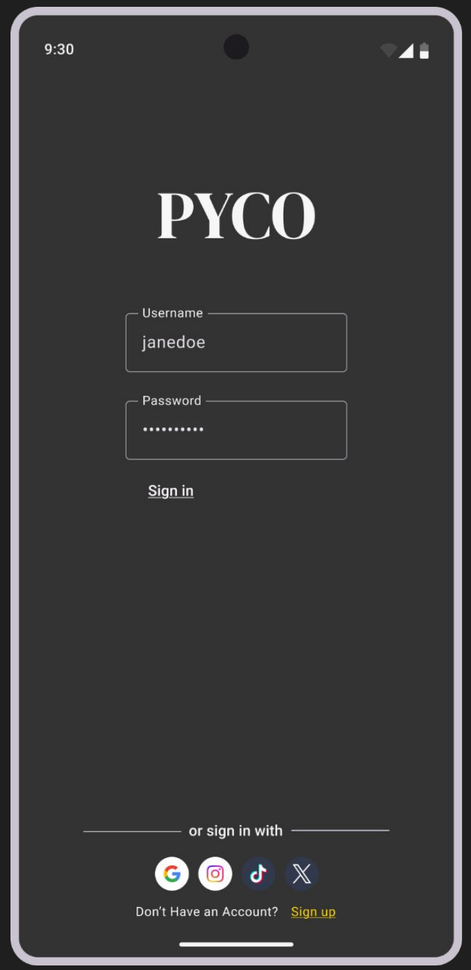

auth with firebase so no need to store passwords.
Edward Marecos
public, private
Akemi Sai
also firebase has photoURL for the pfp
Edward Marecos
user
PK user_id UUID
username VARCHAR
email VARCHAR
account_type ENUM
profile_picture VARCHAR
bio TEXT
follower_count INT
following_count INT
total_likes INT
total_bookmarks INT
total_awards INT
date_joined TIMESTAMP
closet
PK closet_item_id UUID
FK user_id UUID
category VARCHAR
item_name VARCHAR
description TEXT
image_url VARCHAR
brand VARCHAR
color VARCHAR
size VARCHAR
material VARCHAR
pattern VARCHAR
date_added TIMESTAMP
interactions
PK interaction_id UUID
FK user_id UUID
FK outfit_id UUID
type ENUM
timestamp TIMESTAMP
comment_text TEXT
shirt pants etc
Edward Marecos
like, bookmark, comment
Edward Marecos
medals/awards
PK award_id UUID
FK user_id UUID
FK outfit_id UUID
award_type ENUM
date_awarded TIMESTAMP
FK request_id UUID
description TEXT
gold, silver, bronze
Edward Marecos
follows
FK follower_id UUID
FK followed_id UUID
status ENUM
created_at TIMESTAMP
pending, accepted
Akemi Sai
outfit
PK outfit_id UUID
FK user_id UUID
like_count INT
bookmark_count INT
timestamp TIMESTAMP
FK request_id UUID
name VARCHAR
description TEXT
visibility ENUM
items_used FLOAT
like_count INT
bookmark_count INT
requests
PK request_id UUID
FK user_id UUID
event_name VARCHAR
tags JSONB
description TEXT
date_needed DATE
created_at TIMESTAMP
status ENUM
type ENUM
active, expired, completed
Edward Marecos
public, private
Edward Marecos
blocks
FK blocker_id UUID
FK blocked_id UUID
created_at TIMESTAMP
public, private
Edward Marecos
OutfitItems
PK outfit_item_id UUID
FK outfit_id UUID
FK closet_item_id UUID
timestamp TIMESTAMP
RequestItems
PK request_item_id UUID
FK request_id UUID
FK closet_item_id UUID
created_at TIMESTAMP
RequestResponses
PK response_id UUID
FK request_id UUID
FK user_id UUID
response TEXT
response_date TIMESTAMP
status ENUM
accepted, declined
Akemi Sai

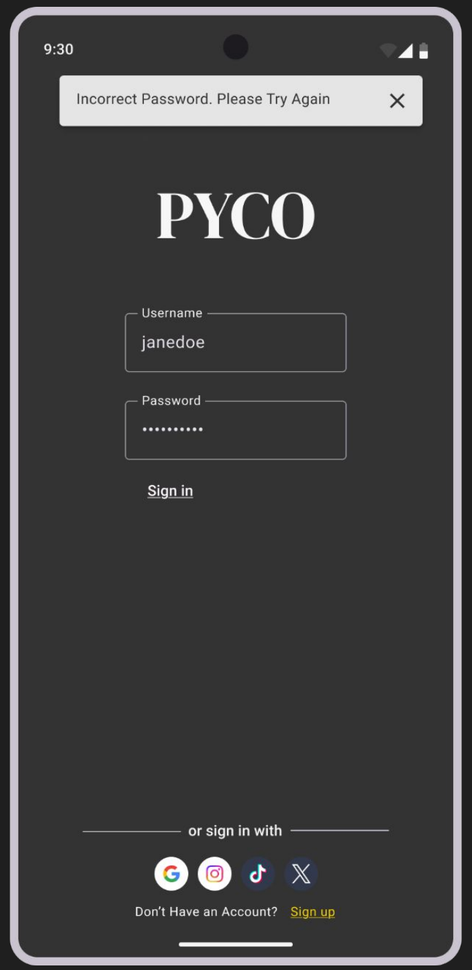
UI Changes since last week

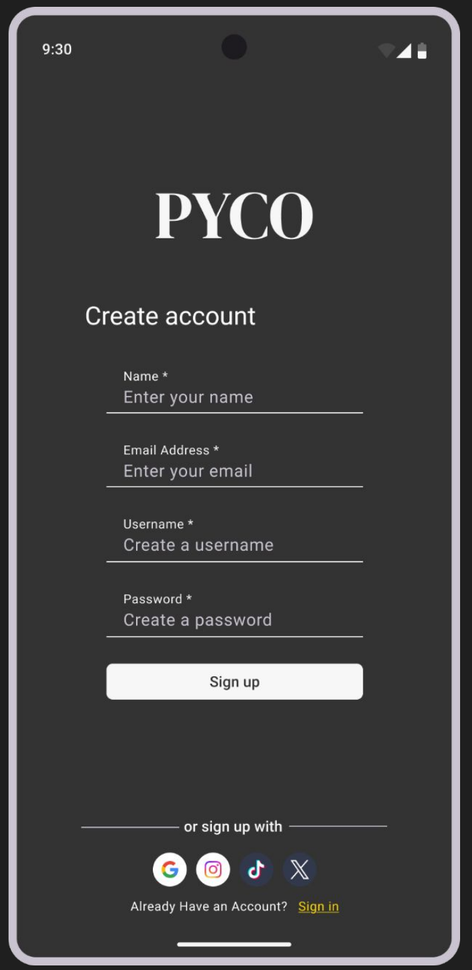
Login screen

- added error for invalid password
- auth with other apps
  - google
  - instagram
  - tiktok
  - X

(apps we felt out target audience is more likely to have)





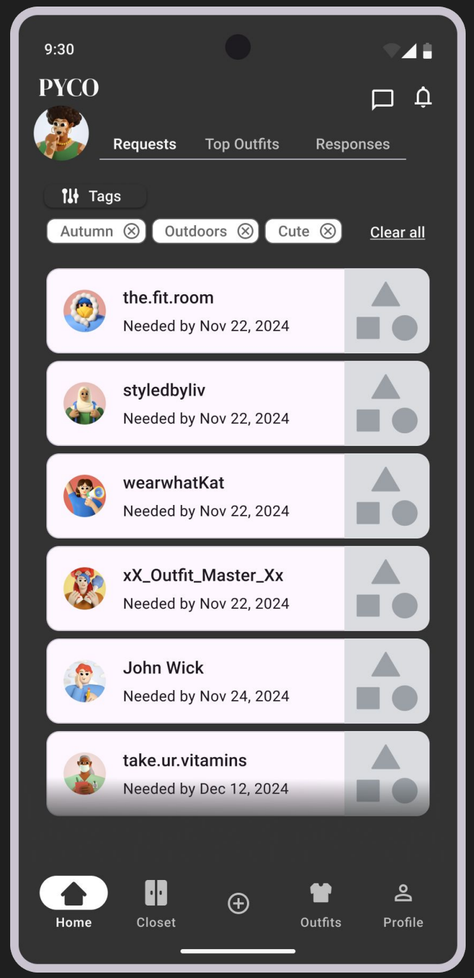


UI Changes since last week

Home screen

- request centered rather than an overwhelming list
- more information per request is visible, users will swipe to view more requests
- the shown tabs were also updated (although not reflected here) and the Top Outfits will display the current most popular outfits as well as a lazy column of completed outfits if people want to scroll
- the responses page will be populated with responses to your requests, as well as your responses to others (separately)

Before

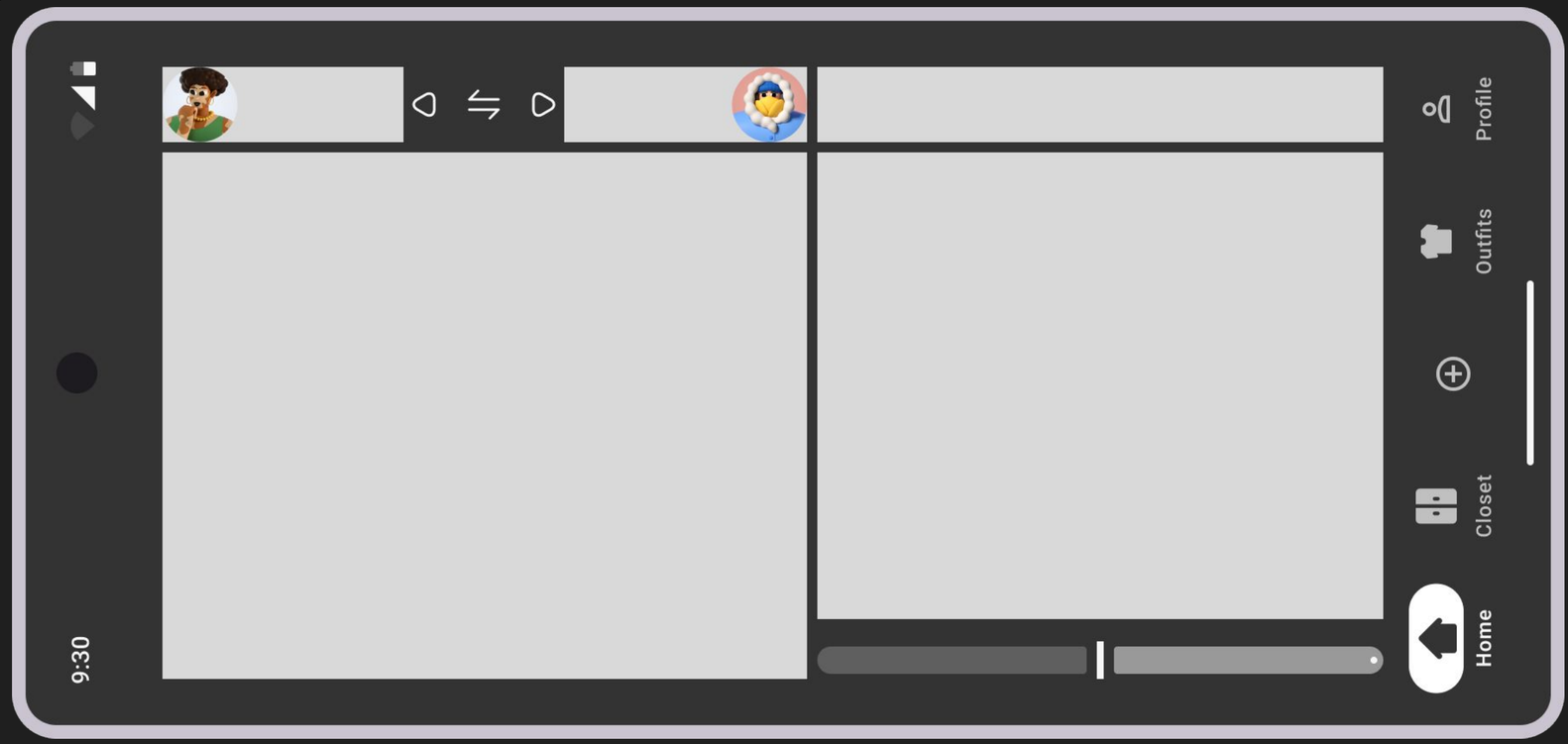


After

UI Changes since last week

Response Page

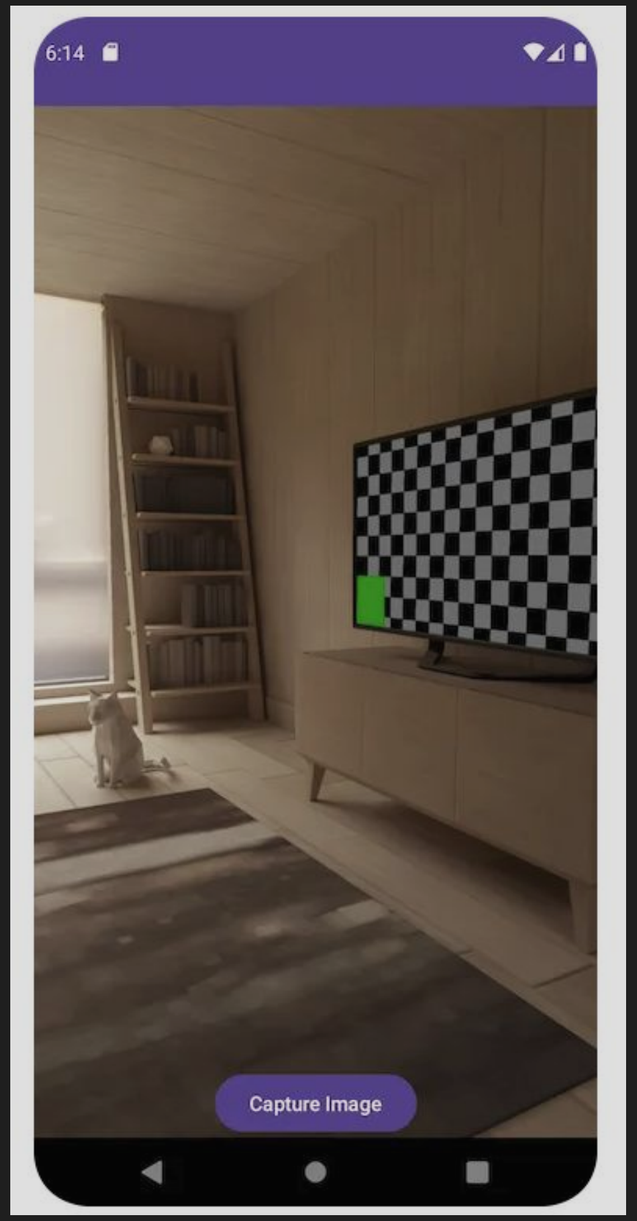
- we would like to implement a portrait version of this, but as of now, landscape is easier to display relevant information. The left large panel will be where you view the outfit you are constructing
- the top right panel will be a slider for viewing the category of clothing
- and the remaining panel will be a display of clothing items in your, or the requester's closet.
- portrait would likely have a similar layout but vertically stacked



UI Changes since last week

Upload Page – Camera

- since we are using the CameraX api the capture screen will look like this~
- the ML kit api will isolate the item photographed
- The CloudVisionApi will then interpret the image to make the process of categorizing the clothing easier for the user
- after the photo is taken users will still be prompted to fill relevant information



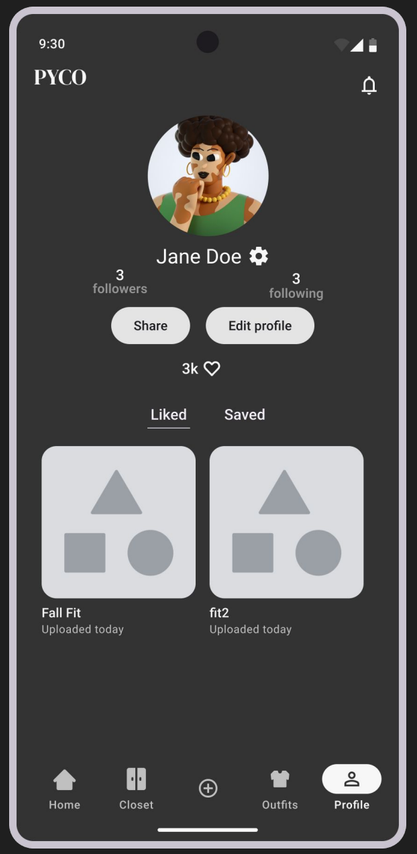
UI Changes since last week

Profile Page

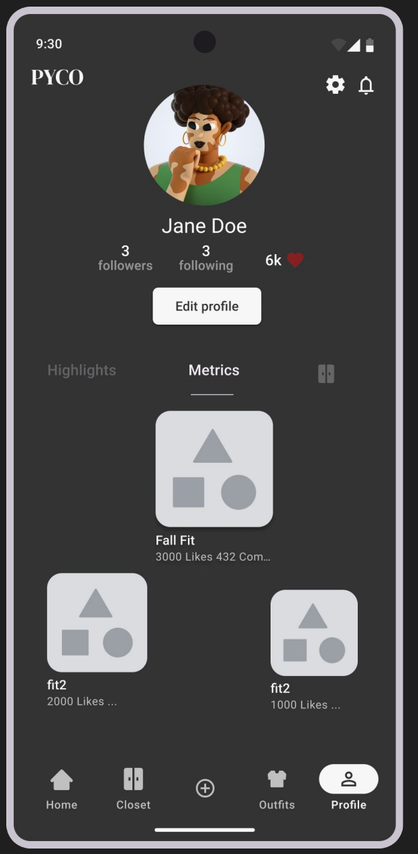
- The gist is the same as last week and we are still deciding on an engaging way to showcase your top outfits.
- The highlights tab will be to demonstrate works you made that you are proud of, regardless of its popularity.
- The wardrobe tab displays your wardrobe (if you have it public)

The remaining UI is consistent with what was presented last week.

Before



Current



View from Else