Team PYCO:
Edward Marecos, Akemi Sai

PYCO

Your Personal Fashion Companion

PYCO Overview

Fashion Accessibility

Aims to make fashion advice accessible to everyone

Creative Platform

Serves as a stage for aspiring designers to build a portfolio and gain recognition

Community Building

Connects individuals looking for fashion inspiration with those who have styling expertise

Long-Term Vision

Potential to evolve into something recognizable for people passionate in fashion.

Tech Stack

CameraX

Image Capture

Remove.bg

Background Removal

Firebase Firestore

Database

Firebase Authentication

Authentication

Jetpack Compose

UI Development
Navigation
State Management

Challenges

Debugging & Logging

Hard to debug and keep track of data when there are so many asynchronous operations like Firestore, CameraX, and background threads. Especially when the View Models are interacting with each other so much.

Databases and Viewmodels

Database Schema changes throughout the development process, a lot trial and error.

Handling Uri and CameraX

Cache management , and URi handling was complicated and took a long time to figure out how to store and fetch the images

Push Notifications

The notifications was hard because we were unsure at first where to place some functions given authentication was not guaranteed at MainActivity.kt

Lessons Learned

API Integration

Learned to handle APIs and manage operations with callbacks

Collaboration

git – a lot of conflicting merges, we improved our communication and planning skills

Databases

Understanding Firestore structure and authentication workflows.

Debugging

Had to get comfortable with understanding gradle file errors, and LogCat for debugging. Many Logs were written.

We mainly utilized Androids LogCat to debug and ensure everything was fetched or uploaded correctly, but towards the end we got Crashlytics up and running and tested it with a crash button (no longer available we vaulted it).

we also have Realtime Analytics to see the traffic our app received, given that it was just us two there was not much to look at.
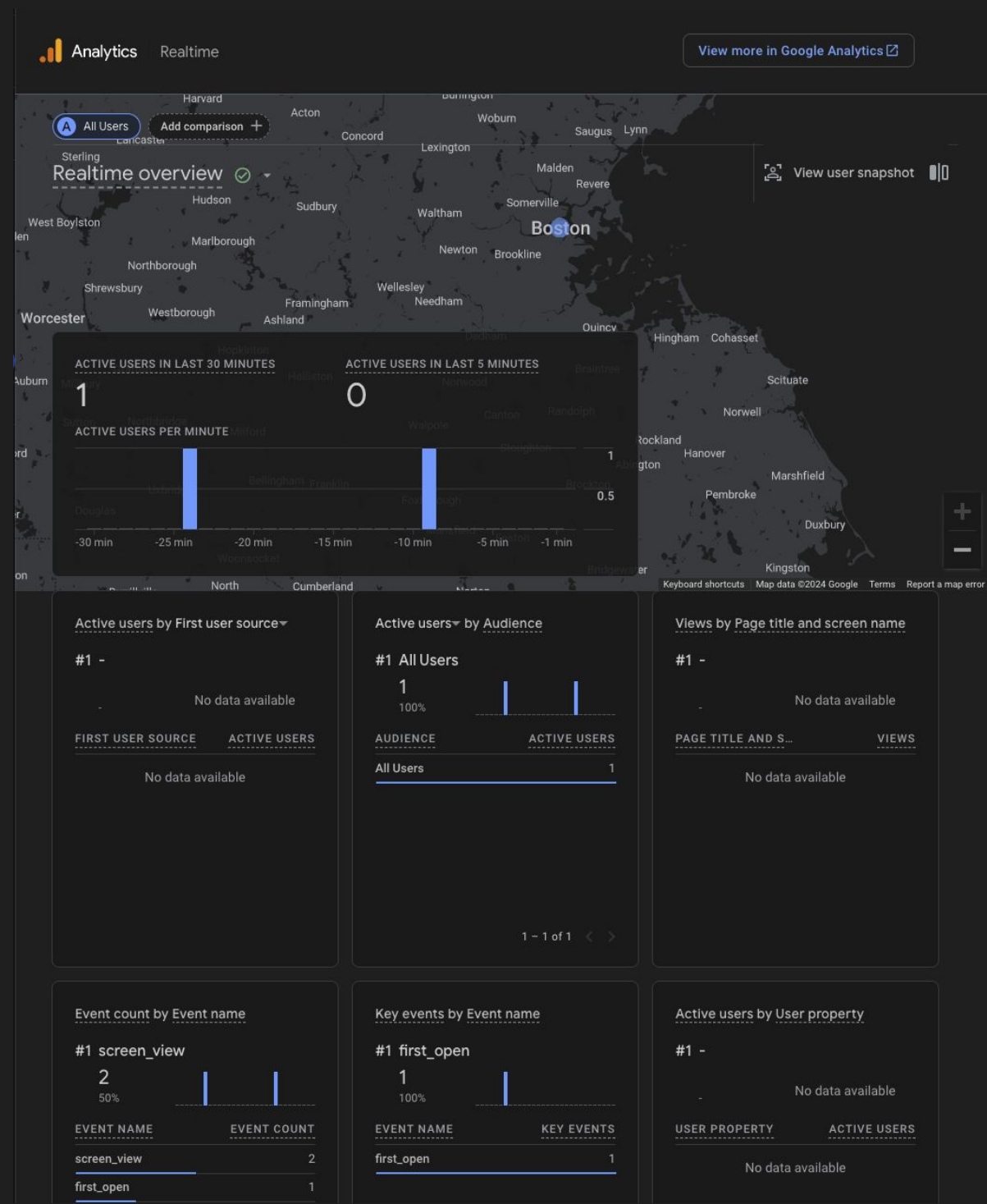
we also bought a phone to test on in areas the emulator fell short. (only 30$)
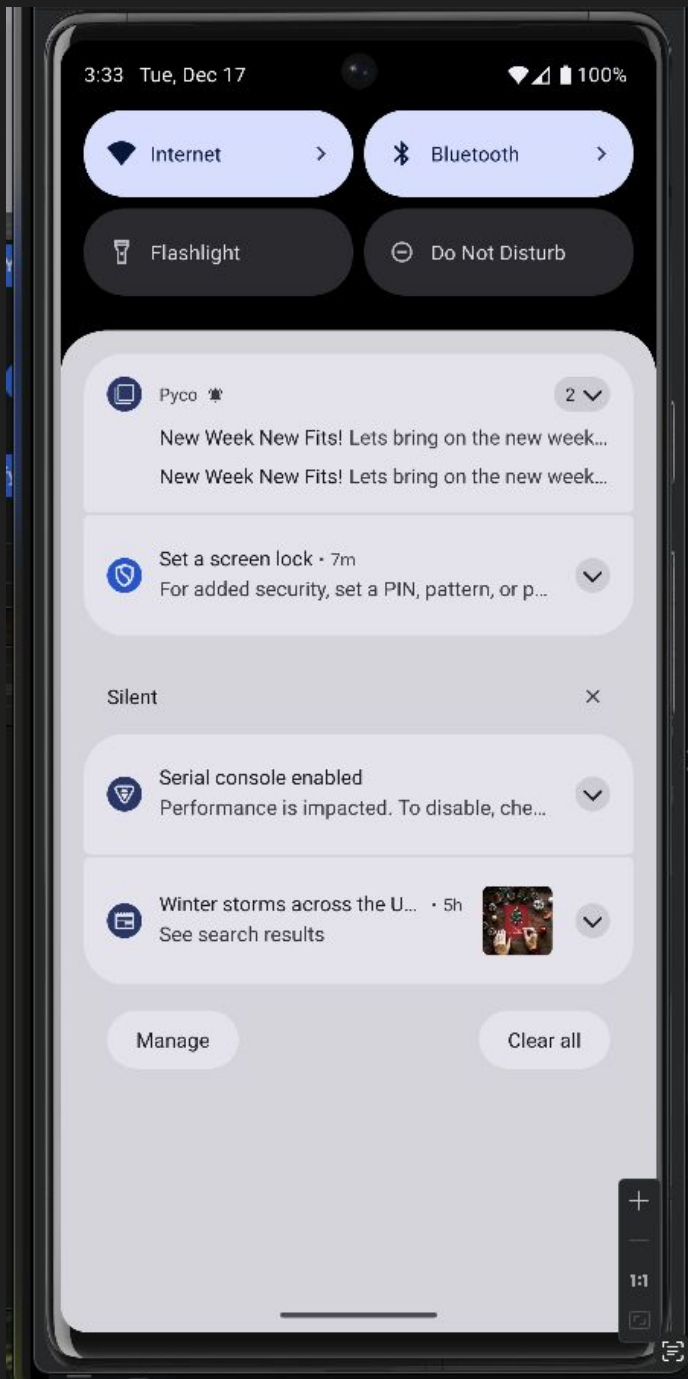
push notifications is also functioning on our app
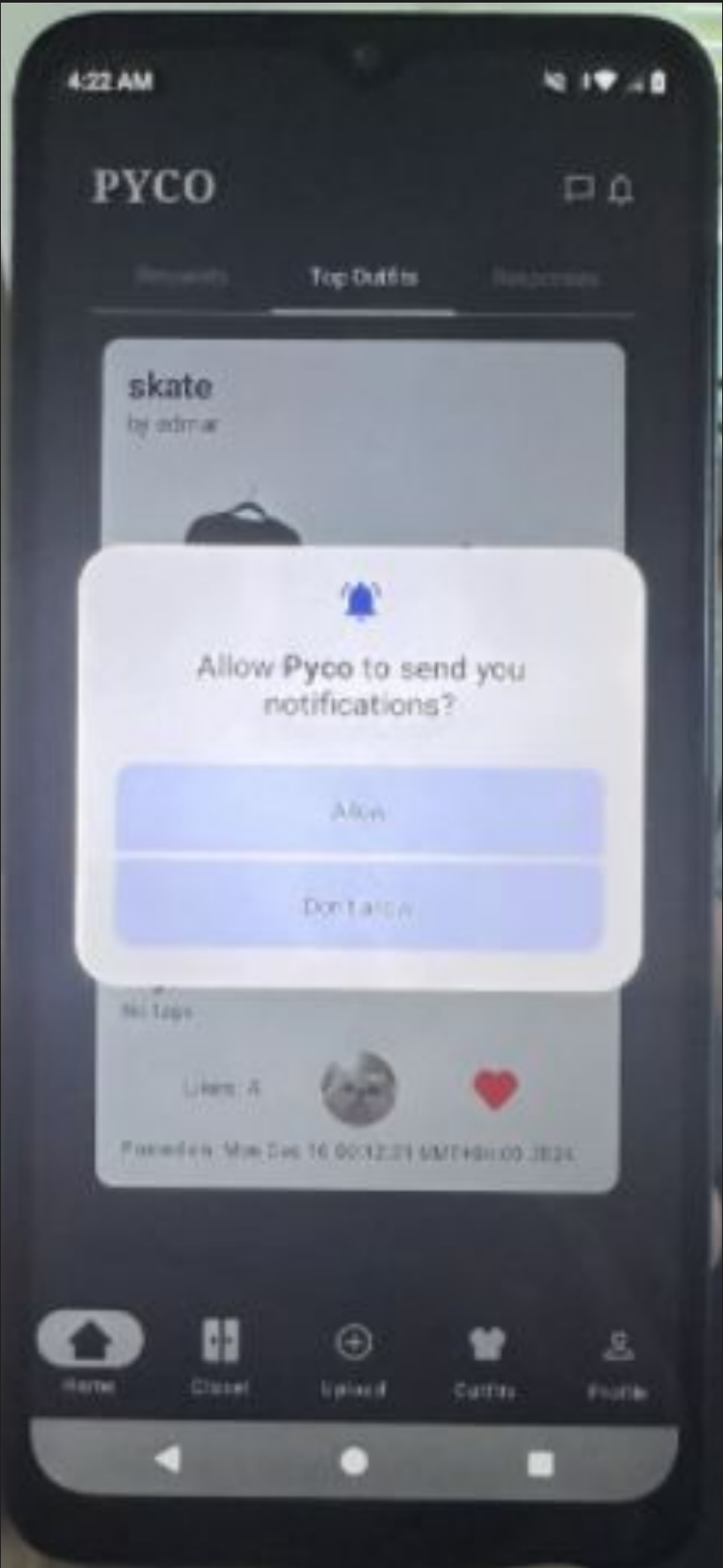
Base features

Google / email authentication

real time data sync with firestore
- **also provides offline functionality**

filtering through requests to access ones

relevant to your interests, or friends!

Clickable items interactive tabs and input fields allow you to really begin to immerse yourself into the app. and there are nice loading spinner animations. and a camera
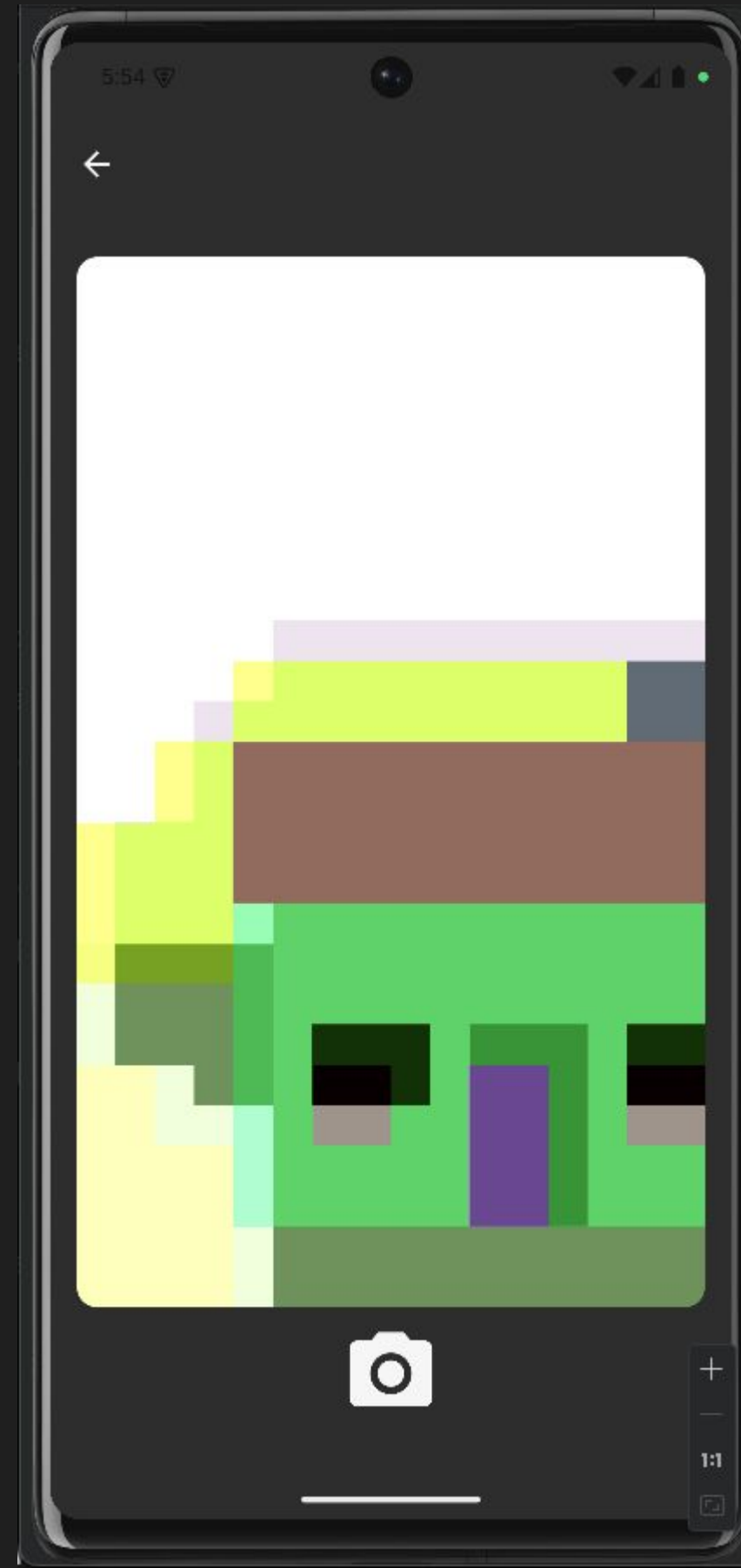
say cheese

we used many api's some of which were for mainly ui such as coil to load images or pagers to slide screens.

we also had apis that did work. Like most if not all of you, we used firebase firestore auth, and we used some of their other services like crashlytics and messaging for notifications.

```kotlin
// firebase
implementation(platform(libs.firebase.bom))
implementation(libs.firebase.auth)
implementation(libs.google.firebase.auth)
implementation(libs.play.services.auth)
implementation(libs.play.services.auth.v2070)
//firestore
implementation(libs.firebase.firestore)
// crashlytics
implementation(libs.firebase.crashlytics)
implementation(libs.firebase.analytics)
// messaging
implementation(libs.firebase.messaging)


// camerax
// CameraX core library using the camera2 implementation
implementation(libs.androidx.camera.core.v150alpha03)
implementation(libs.androidx.camera.camera2.v150alpha03)
// If you want to additionally use the CameraX Lifecycle library
implementation(libs.androidx.camera.lifecycle.v150alpha03)
// If you want to additionally use the CameraX View class
implementation(libs.androidx.camera.view.v150alpha03)
// If you want to additionally add CameraX ML Kit Vision Integration
implementation(libs.androidx.camera.mlkit.vision)
// If you want to additionally use the CameraX Extensions library
implementation(libs.androidx.camera.extensions.v150alpha03)


// coil for image loading
implementation(libs.coil.compose)
// some icons for ui
implementation(libs.androidx.material.icons.extended)

// accompanist pager library for the horizontal pager on outfit feeds // among other screen like followers etc
implementation(libs.androidx.foundation)


// guava
implementation(libs.guava)


// Firebase App Check
implementation(libs.firebase.appcheck.playintegrity)
```

There were a few features we didn't get to complete, auth with other services besides google

the outfit creation screen is underwhelming in terms of ui, and the account screen has also seen better days, but functionality wise we hit many of our markers.

We did not include local messaging and notifications like we initially planned as well.

time for a demo?