# SocialData: Design Implementation & Challenges

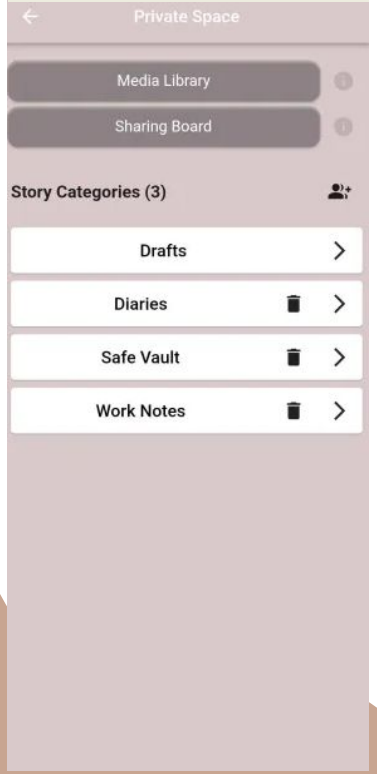
Client: Wensheng Zhang
Group 1: Anton Kordick, Brian Kurz, Jackson Smith, Marshall Moats, Rodrigo Santamaria

# Our Project

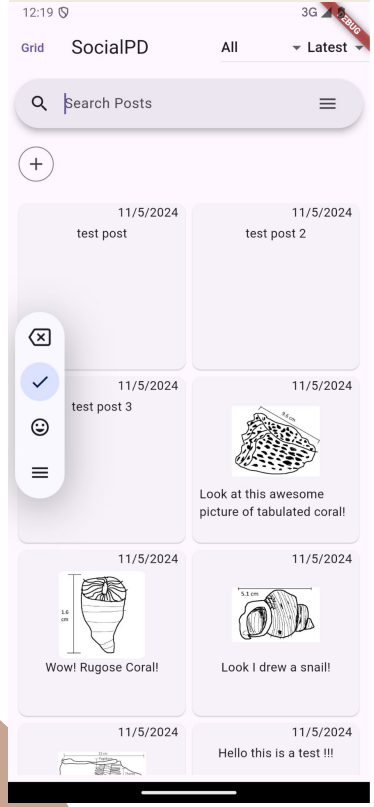
Recap summary of our project:

- Motivation is to enable users to have centralized, independent control of the data shared on social media platforms (e.g. images and videos)
- Our project is to create an application that:
  - is available on multiple platforms (Android, iOS)
  - lets users to connect accounts from social media platforms
  - fetch posts made on connected accounts
  - stores files locally
  - lets users view files in one place

Design of prototype given by Client



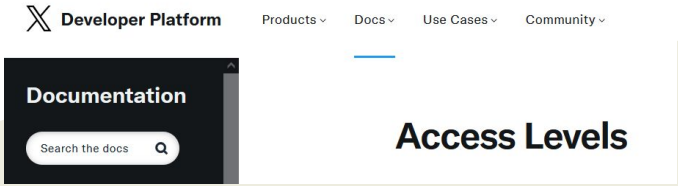

Current design



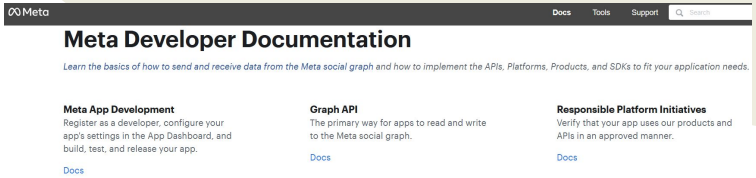

# App Layout / UI

- Not much instruction from client on UI design, other than having functionality
  - More of a gallery-like app than typical media platform
- Mostly the same layout since Demo 1
- Looking to other gallery and media apps for inspiration
- Design choices:
  - List/grid view
  - Tags to filter by platform, tag search
- Required to use Flutter, Dart for development
  - Learning curve for all members

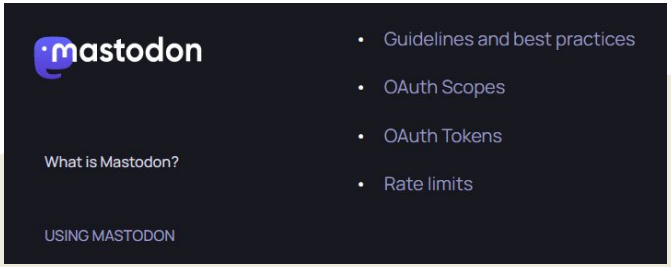
# Linking Accounts / API Calls

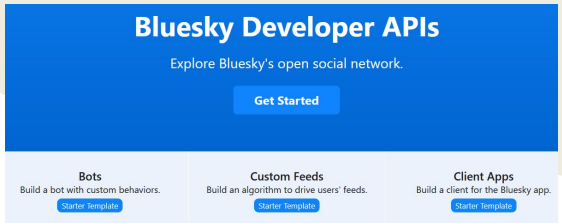



- Started out with Facebook, Instagram, and X as main integration focuses as determined by client
- Problems with using X & Meta API
  - Paid tiers of access
  - Outdated information



# Linking Accounts / API Calls




- Had discussion with client about shifting focus to decentralized media platforms
- Shifted focus to Mastodon & BlueSky

# File Storage

- All files are pulled from various social sites and stored locally on the user's device in the app documents directory
- Means we have to compare what files we have stored vs the ones retrieved from the social networks
    - Need a diff algorithm in case a user edits a post
- File metadata is stored in our local database as sqlite tables
- All files are intended to be stored analogous to our app (don't depend on our app)

# Frontend Technologies




- Riverpod for state management
  - State management is a pain
  - Riverpod is the successor to provider
  - Allows for our state to notify components when they change
- Using Isar for our local database
  - ORM using SQLite engine
  - Allows for easy linking between collections
  - Allows for easy indexing and conflict resolution
  - Allows for easy filtering/searching db
  - Can watch queries or collections for automatic UI updates
- Path_provider
  - Allows us to get directories regardless of platform
  - Can store files locally in a place that our users can find
- Http
  - Most basic
  - Allows for http requests with headers and parameters



# Current Prototype (prerecorded demo)

Video

# Expected Progress by Demo 3

- Better platform support–currently only Mastodon is working as intended
- Expired posts are deleted when the app is opened
- Posting and deleting posts on social media platforms using this app
- Add accounts (follow/add friends) on connected platforms
- More forms of content retrieved by the app (statuses, likes, DMs, etc.)
- Filtering and sorting by media type
- Searching by hashtags
- Users can lock their media behind a password
- Multiple languages
- Users can see what data is being shared on different platforms
- Users can revoke data sharing permissions for specific platforms
- Users can store their data on the cloud rather than locally (Google Drive, AWS, OneDrive)

# Thanks for listening!