Bulletin Magazine Mobile App

Background

Bulletin Magazine is currently distributed as physical copies and as PDF's on thebulletinmag.com, a site hosted using Google Sites. The PDF's are stored in a Google Drive folder and served on the website using the embedded PDF viewer. This experience works well for desktop users, but is a poor experience for mobile users.

Goals

- Provide students, teachers, and parents with a mobile-friendly viewer for Bulletin Magazine
- Two components: Server (host pdfs) and client (Android/iOS apps)
- Project scope should be bounded as something achievable by computer science club members.

Serving + Hosting Infrastructure

- Must facilitate the hosting and transfer of Bulletin PDF's
- Must be easy for a non-technical authenticated user (e.g. Ms. McCluskey or student staff of Bulletin Magazine) to upload new PDF's as new issues of the magazine are created.
- Must run under free (or very cheap) tier of hosting provider.
    - There is no allocated budget for this project.
    - The hosting provider must **prevent** any over-use which might cause project to be billed (e.g. usage beyond free tier).
- Must require little-to-no ongoing maintenance.
    - Students will be working on the app, but students eventually graduate.
    - Maintaining a dedicated server (installing updates, etc) should be considered beyond the scope of what is desired for this project.

Client App

- Must list available issues PDF's
- Must allow users to download and view PDF's
    - Should allow users to view PDF's within app

- Should support both Android and iOS
    - Initial release may support only one platform

Nice-to-have

- ✓ Project could be open source to allow participating student(s) to have evidence of their work to include on a resume or portfolio.
- Client app could have an opt-in setting to notify users of new issues of the magazine.

Suggested Path For Building The App

- Firebase for server component
    - Won't bill when going over usage
    - Only option found so far that checks all requirements.
- (Suggestion) React Native for client app
    - Very fast for building simple apps
    - Shared codebase between iOS and Android
    - May be a less ideal experience than two dedicated mobile apps.

1. Setup hosting for source code of project

- Plenty of options for this
    - https://github.com
        - Most popular for hosting code
    - https://bitbucket.org
    - https://gitlab.com
- Try to find someone who can review pull requests for the project!

2. Build a proof of concept PDF viewer for either Android or iOS

- See 'Client App Details' section below
- Building the PDF viewers will probably take the majority of the time for this project.
- Suggestion: focus on one platform first

3. Create Firebase project for serving a PDF

- Will eventually need to be something owned/managed by BullETin magazine staff, but not needed for initial proof of concept
    - Create a free account and worry about it later
- Create a new storage bucket
    - **Make sure only authenticated users can update storage bucket**
    - A firebase cloud function may be required to get a list of available files in the 'bucket'

4. Publish Minimum Viable Product (MVP)

- Initial release can support just one platform
- Get user feedback on initial release!

5. Cleanup and publish final version

- After building app for first platform, second platform should be much faster.
- Make sure Firebase project is now properly set up
    - ✓ Anonymous reads of the storage bucket should be allowed
        - Anonymous 'writes' should be disallowed
    - Ensure journalism students have access to Firebase project

Related Information:

Hosting / Storage Options Considered

- Hosting and Storage (as described in this document) are two distinct needs:
    - **Storage**: Uploads and stores the data. Served to hosting provider.
        - not all storage providers allow serving directly to clients/apps).
    - **Hosting**: Serve content from storage provider
        - Allows for authenticated access of storage
- Ideally the same provider is used for both to reduce complexity

Firebase (Hosting + Storage)

Firebase is a cloud platform focused on providing back-end infrastructure (hosting/storage) for mobile apps. Firebase is partially backed by Google Cloud Platform, but appears to have additional limits in place to prevent you from exceeding free usage.

Firebase has a generous free plan (named 'Spark') which allows for basic usage from a given project. This include a small amount of storage and a small number of operations per month.

The spark plan should provide more than adequate usage (both storing and serving) for this project.

There is a tool for uploading files from the web into firebase storage

Pricing

- See: https://firebase.google.com/pricing/
    - Note: they have a section named "hosting" which is not applicable to this project.
- Free tier: for "Storage" under "Spark"
    - 5 GB stored
    - 1 GB downloaded / day
- Note: Firebase explicitly does not offer any further discounts for education / nonprofit usage.

Handling Overages

See: https://firebase.google.com/support/faq/#pricing

- What happens if I exceed Spark or Flame plan upload, download or storage limits for Cloud Storage?

- TLDR: they disable the app temporarily rather than charge you.
- When you exceed limits for Cloud Storage in a project on the Spark or Flame plan, the result depends on the type of limit that you exceed:
    - If you exceed the GB stored limit, you will not be able to store any more data in that project unless you remove some of the data stored or upgrade to a plan that provides more storage space, or unlimited storage space.
    - If you exceed the GB downloaded limit, your app will not be able to download more data until the next day (starting at midnight, US Pacific

Time), unless you upgrade to a plan with less restrictive limits, or with no limits.

  - If you exceed the upload or download operations limit, your app will not be able to upload or download more data until the next day (starting at midnight, US Pacific Time), unless you upgrade to a plan with less restrictive limits, or with no limits.

Risks

- Firebase may reduce their free plan in the future.
  - This is a risk with any free offering
- If more than 1GB is download in a given day, the app will be unable to download until the next calendar day.
  - If someone were to 'abuse' the app and repeatedly download these files, they could prevent the app from working for a day by causing it to run out of quota. Cache? Client Limit?

(Requires Storage) Heroku Free/Hobby tier (Hosting only)

- Heroku has a generous free tier for hosting web applications
  - See: https://devcenter.heroku.com/changelog-items/907
  - TLDR: You are allocated up to 1000 hours of runtime per month per account for free apps.
    - This allows you to run ~1.3 web apps per account per month for free.
- Problem: Heroku does not appear to also provide 'storage'.
  - A common solution to this is to store content in Amazon's S3, but S3 does not have a free tier that lasts longer than a year.

Risks

- When going over the allocated number of hours per month, Heroku will 'sleep' your application until you have available hours again.
  - If this were to happen (it shouldn't if 1000 hours/month are available), the app will be unusable for up to a month.
- Heroku has changed their free tier a couple times in the past few years and there is the risk they may again reduce it.

shouldn't happen normally

(Dangerous) Google Cloud Storage Always Free Tier (Hosting + Storage)

- Google Cloud Platform has an 'Always free' plan which allows for basic usage for free forever.
    - Note: requires a credit card to sign up.
- Google Cloud Storage allows for 5 GB-months of storage per month for free.
    - A storage bucket could be created which allows for public access (preventing the need for other hosting)
    - Only 1 GB of download traffic is provided for per month
        - After 1 GB of usage, account is billed $0.12/GB
            - While it is unlikely for someone to abuse this, it would certainly be possible for someone on a home connection to download ~300GB/hour

Risks

- GCP does not **cap** usage when you go over your budget and merely alerts you.
    - Because of this, it is possible that someone could repeatedly download these files and rack up charges.
    - This is different from Firebase which will instead temporarily turn off your app.

(Infeasible) Using google drive API (Storage)

- PDF's are currently stored in a folder on Google Drive so serving the files from there might be ideal
- The Google Drive API unfortunately does **not** allow for anonymous access of external Google Drive folders, so this does not appear to work.
    - See: https://developers.google.com/drive/api/v3/about-auth
    - PDF's require end-user's to authenticate with OAuth:
    - Your application must use OAuth 2.0 to authorize requests. No other authorization protocols are supported. If your application uses Google Sign-In, some aspects of authorization are handled for you.
    - Users are meant to authenticate their personal accounts:
    - Google displays a consent screen to the user, asking them to authorize your application to request some of their data.

Risks

- None, because it doesn't work for our purposes lol

Client App Details

- Components:
    1. Listing PDF's
    2. Fetching from server
        - (Optional) probably want a library for fetching from whatever is serving the PDF's
            - Depends on Hosting / Storage chosen
    3. Viewing PDF
        - Definitely do not want to build this yourself
        - Rely on a library or builtin support in the platform
- Could be built as native app or using React Native

React Native App

- React Native is a popular project for quickly developing basic cross-platform (Android + iOS) apps from a single codebase
- Requires some initial setup.
- React native PDF libraries:
    - https://github.com/wonday/react-native-pdf ☆
        - 500 stars

Native Apps

- Would require app to be built twice
    - App is simple enough that writing the app twice may not take too long
- Android: Java or Kotlin
    - PDF Libraries:
        - https://github.com/barteksc/AndroidPdfViewer
            - 4.3k stars
- iOS: Objective-C or swift
    - PDF Libraries:
        - https://developer.apple.com/documentation/pdfkit
            - built into iOS (version 11 and higher)!

Requirements to publish apps

Android

- 64-bit required April, 2019 as of changing plan by play Store
- Requires 1 time fee of 25 dollars for a developer account
- Low enough cost to be paid for.

iOS

- Requires $99/year developer account
  - Probably too expensive for Bulletin mag
- Free for Education / non-profits
  - https://developer.apple.com/support/membership-fee-waiver/
  - requires you provide extra information
    - D-U-N-S number
    - Tax ID
  - Ms McCluskey is investigating this

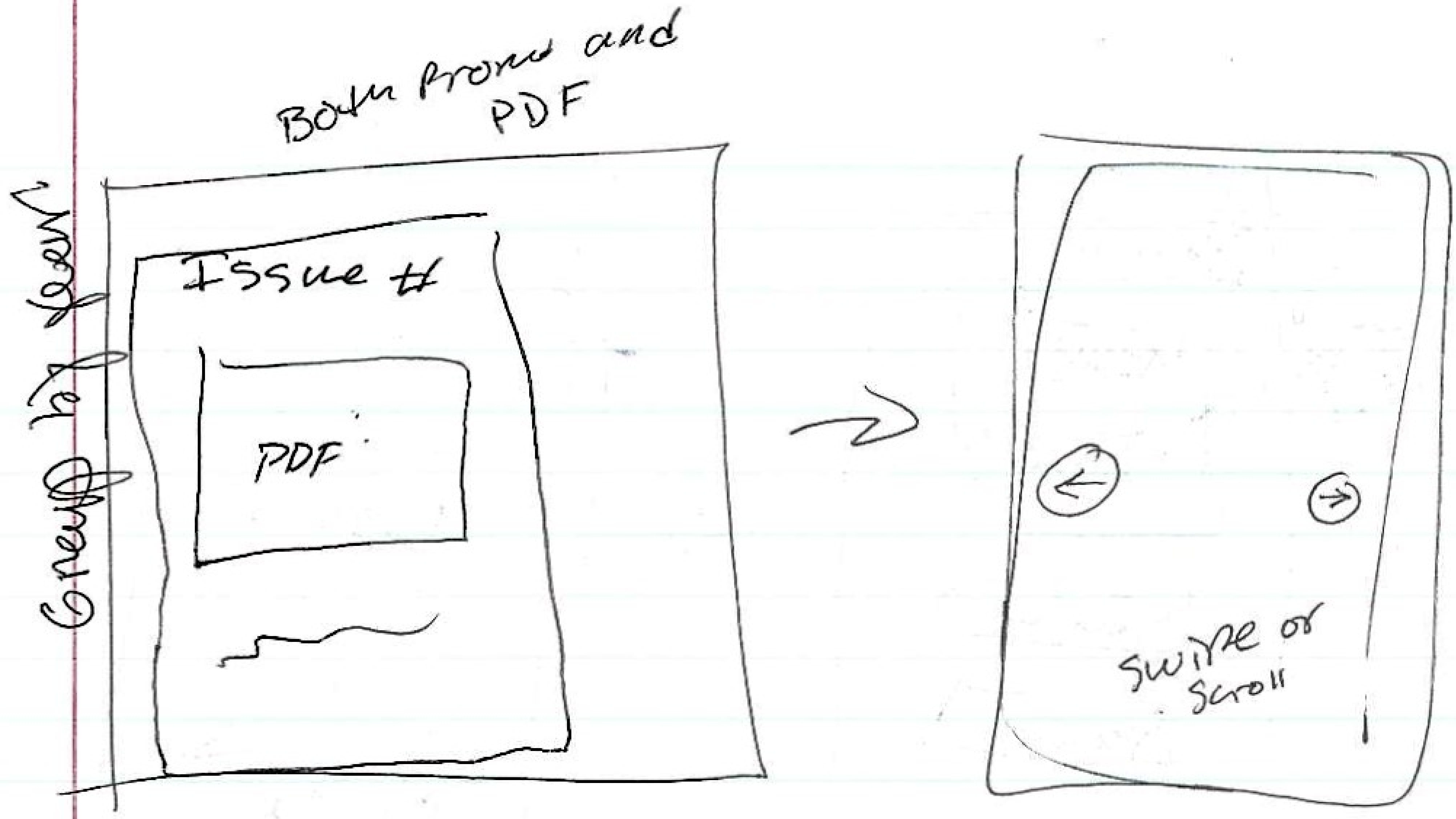
~~iOS - $100 /year~~

What platforms? ~~Android - $~~

Bulletin magazine app requirements

relies?

1. Google drive api client - Will look into, although highly unlikely
    1. App is deployed with an api key that allows access to a specific folder in shared Google Drive — What kind of folder? (Computer, GDrive, ect.)
2. App is able to list files in the folder and sort them by date
    1. Top of the list is most recently modified issue
    2. We can assume that issues will not be republished after a newer version comes out
    3. This list can serve as the homepage for the app
3. App can open both PDFs and videos in the list
    1. Nice to have: PDFs are viewed one page at a time (full screen)
    2. Nice to have: PDF page navigation is horizontal (to swipe left and right) because this is a magazine
        1. Scrolling is ok but not ideal
4. Image for app icon will be decided later. My students will either make one or the ETHS bull will be used
5. Thank you :) let me know if you need anything from me or have any questions.

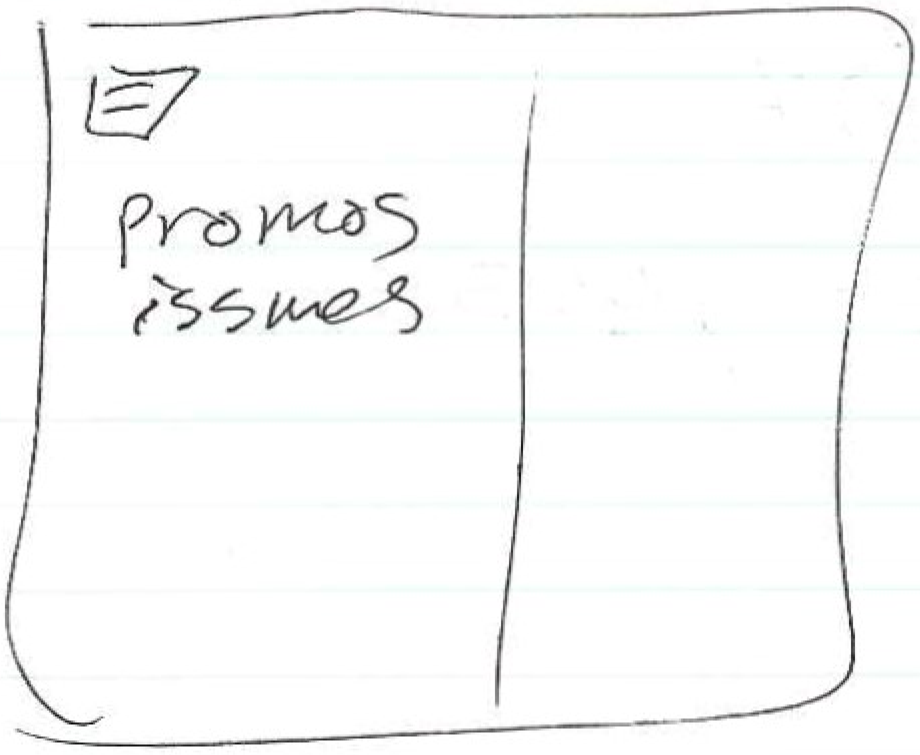


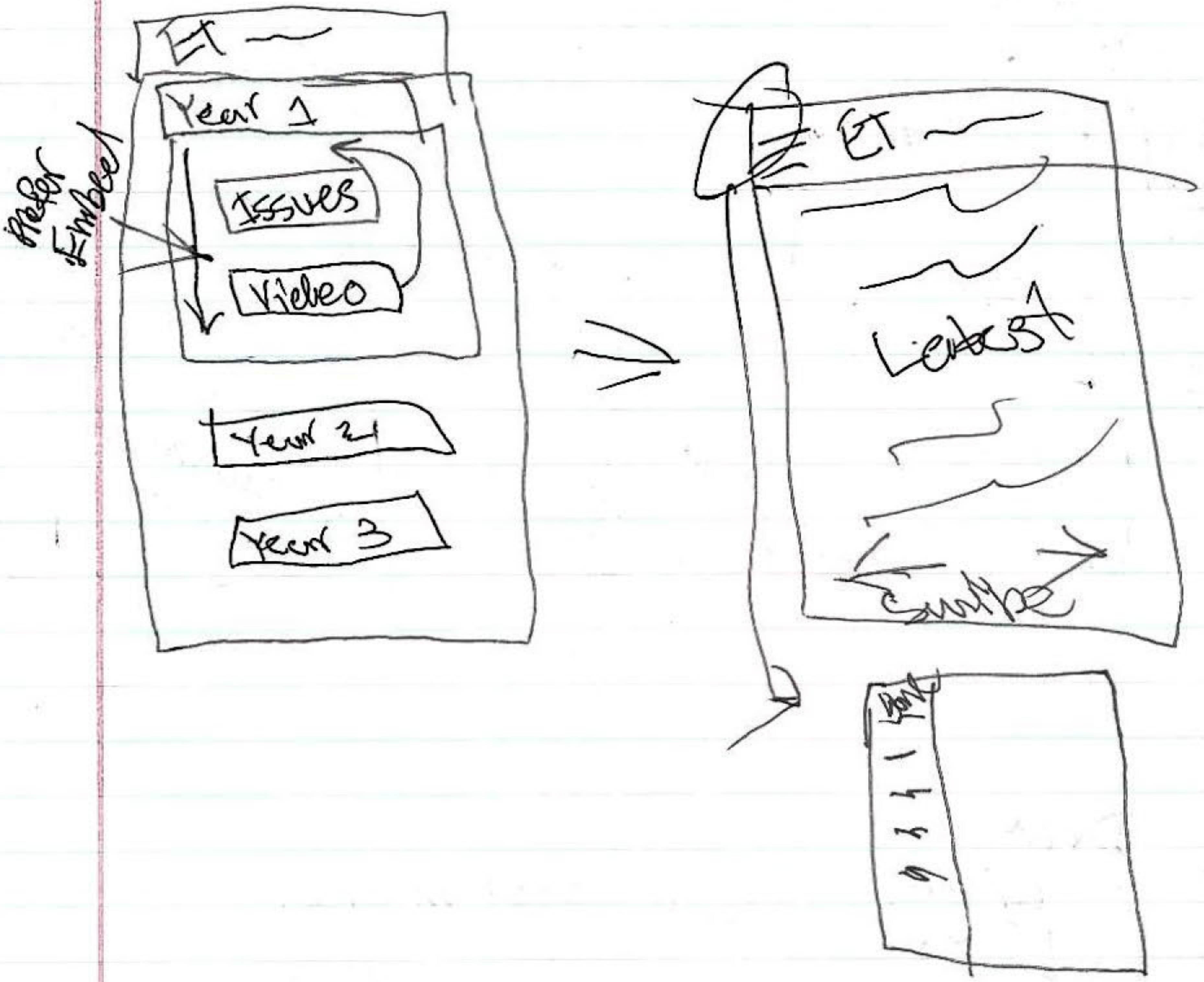
Drive
□ yr 1
□ y2
□ .

→ post in order

Bottom to top
1st Recent

Fallback




ET
Year 1
Issues
Video
prefer Embed
Year 2
Year 3
ET
Latest
Swipe
Done

[redacted]