Designing for Instapath

Daniel Wei

Instapath is a Y-Combinator-backed startup based in the Texas Medical Center. Their primary product offering is a physical hardware device called Luci, a patented microscopic imaging system which allows doctors to get pathology scan results in minutes, not days.

I first joined Instapath in 2022, and have been working with them on and off over the years. It was only in the Summer of 2023 that I had the opportunity to upgrade my role to that of Product Engineering. Along with the title came full ownership of their primary software offering: the Luciviewer — a web application that, when paired with their Luciviewer, allows users to upload, view, organize, inspect, analyze, and recolor scans with ease. They developed this own software offering out of frustrations with the existing industry standard platform for viewing scans: the Digital Slide Archive. My job: to redesign, and then reimplement the entire software program.

This exhaustive story aims to illustrate my UX design process during this ambitious journey, especially for three core user experiences, which I will present in case study format.

Table of Contents

General Design Constraints

I was limited by the following **constraints:**

- Time. My work was limited to a single summer. The entire design process had to be compressed correspondingly.
- Technical ability. Since I was also the primary (read: only) frontend software engineer on the team, I was designing an experience that I would also have to code out by the end of the summer.

Target Users

The following information is gathered based primarily on company analytics. As a B2B (business-to-business) enterprise, Instapath conducts most of its business with top-level executives in hospital operations.

Some **characteristics** of our target demographic:

- Doctors and nurses of the medical profession. Occasional technical or supporting staff, though this is rare.
- Primarily middle-aged or above, due to the nature of the medical field. This is not an overwhelmingly tech-literate audience.
- Conservative attitudes toward the adoption of technology. As a consequence of a need for stability and reliability, as well as personal and cultural factors, this demographic rarely picks up on new technology quickly.
- Busy. Doctors often need to switch contexts between quick patient visits frequently, without losing their train of thought or ability to function at an exceedingly high level.

Consider the following **user persona:**

> *John is a 42-year-old doctor based out of Houston. Every day, he commutes an hour from his suburban home on the west side of the city to work in the Texas Medical Center, where he logs and files 6 or more patient scans within an hour, adding to an endless backlog of scans. With a prescription for blue-light glasses himself, he's experiencing eye strain from making out small details on individual cells as well — not to mention the fact that the artificial coloring of a scan might not always be the best suited for contrast or unbiased analysis. Working on a low-powered, small computer mounted on the wall running Windows XP, he makes paper notes and writes prescriptions with one hand, and navigates the screen using a mouse with the other. He is hopeful that Instapath will change the way he does patient scans, but he's skeptical that any solution they come up with will fit the technical constraints and fast-paced nature of the hospital. He finds technologies like Instapath cool in theory, but often unreliable in practice, and is hesitant about the hospital's impending switch to using Instapath for all their imaging needs.*

This imposed the following **additional constraints** on my design choices:

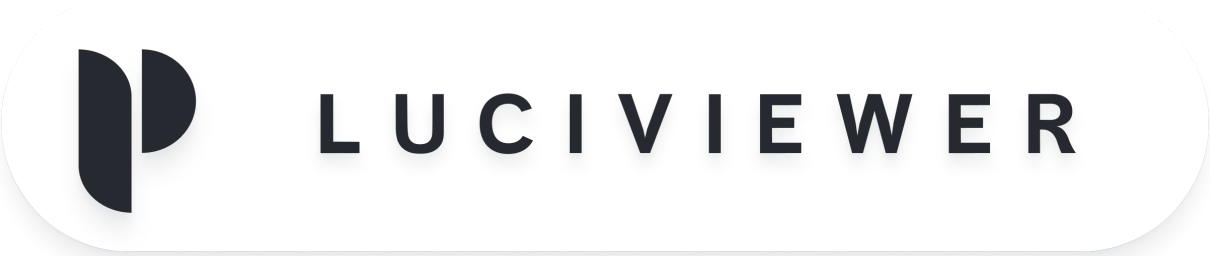
- The design must be intuitive and *simple by default and powerful as needed*. This kind of "revealed" functionality is essential given the tech literacy of my target user.

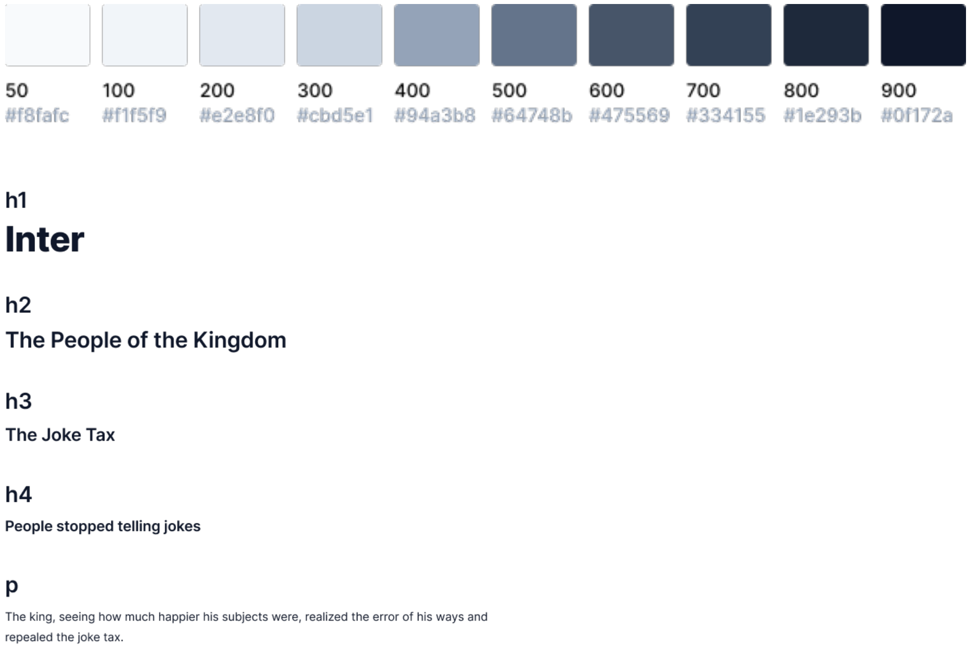
- The design must make use of a *limited FOV* (field-of-view). The more screen real-estate dedicated to the scan itself, the better — especially on outdated, low-resolution, small computers.
- The design must be friendly for use with *only one hand.*
- The design must assist in reducing color bias. In addition to the obvious parts (i.e. developing a feature for manually recoloring the image), the visuals should use a *neutral color scheme (greyscale)* in order to reduce bias towards certain colors in image-viewing!

Branding & Design Kit

As a consequence of the constraints (see Color Bias constraint), I chose to design a greyscale variant of the general Instapath branding for the Luciviewer.

In order to speed up the design and development process (see Time constraint), I used @shadcn's fantastic component library, available both for React and on Figma, to compose my designs. Arbitrary branding decisions are below:





50	100	200	300	400	500	600	700	800	900
#f8fafc	#f1f5f9	#e2e8f0	#cbd5e1	#94a3b8	#64748b	#475569	#334155	#1e293b	#0f172a

h1

Inter

h2

The People of the Kingdom

h3

The Joke Tax

h4

People stopped telling jokes

p

The king, seeing how much happier his subjects were, realized the error of his ways and repealed the joke tax.

Case Study 1: The Image Library

The first core capability that comes to mind is the image library. A reasonably large client (eg. a mid-sized hospital in the Texas Medical Center) could easily upload hundreds of images a day or more. For a doctor to pull up someone's scans on-file within a reasonable time frame requires a comparatively powerful interface for finding specific images or groups of images.

Parts

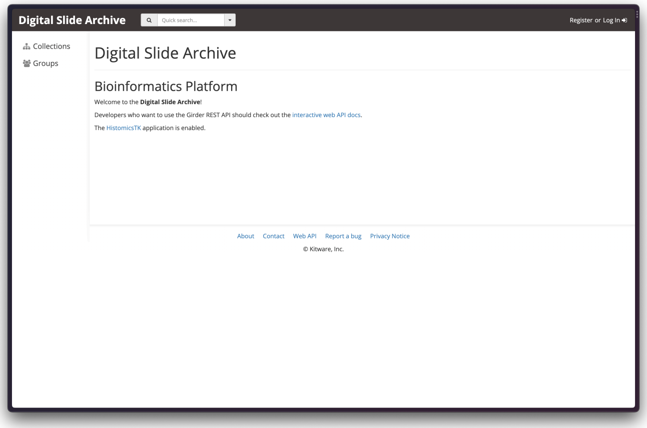
1. Research
   a. Competitor Analysis
   b. Observations
      i. Previous User Journey
      ii. Pain Points
2. Redesign Goals
   a. Measuring Success: KPIs
   b. Hypotheses
3. Brainstorming
   a. Speculation/Moodboarding
   b. Impact v. Effort Matrices
   c. Lo-Fis
4. Finalization
   a. User Feedback
   b. Selecting a flow
   c. Hi-Fis

e. Conclusion
   a. Unsolved Problems and Next Steps

Part 1: Research

Competitor Analysis

Our only competitor in this field is the industry standard, the open-source Digital Slide Archive (DSA) project.



Digital Slide Archive

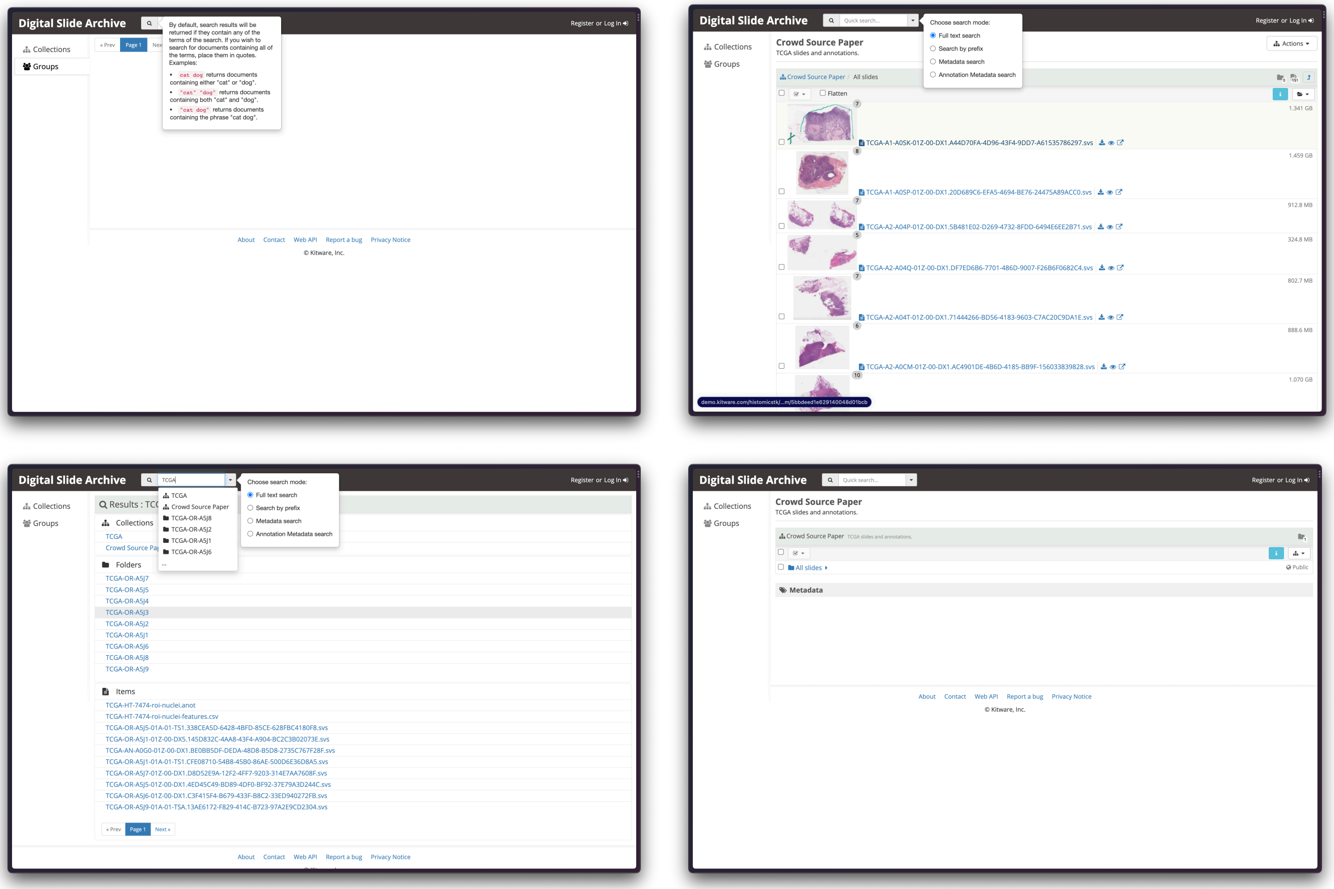
Bioinformatics Platform

Welcome to the **Digital Slide Archive**!

Developers who want to use the Girder REST API should check out the interactive web API docs.

The HistomicsTK application is enabled.

© Kitware, Inc.



By default, search results will be returned if they contain any of the terms of the search. If you wish to search for documents containing all of the terms, place them in quotes. Examples:

- `cat dog` returns documents containing either "cat" or "dog".
- `"cat" "dog"` returns documents containing both "cat" and "dog".
- `"cat dog"` returns documents containing the phrase "cat dog".

TCGA-A1-A0SK-01Z-00-DX1.A44D70FA-4D96-43F4-9DD7-A61535786297.svs 1.341 GB
TCGA-A1-A0SP-01Z-00-DX1.20D689C6-EFA5-4694-BE76-24475A89ACC0.svs 1.459 GB
TCGA-A2-A04P-01Z-00-DX1.5B481E02-D269-4732-8FDD-6494E6EE2B71.svs 912.8 MB
TCGA-A2-A04Q-01Z-00-DX1.DF7ED6B6-7701-486D-9007-F26B6F0682C4.svs 324.8 MB
TCGA-A2-A04T-01Z-00-DX1.71444266-BD56-4183-9603-C7AC20C9DA1E.svs 802.7 MB
TCGA-A2-A0CM-01Z-00-DX1.AC4901DE-4B6D-4185-BB9F-156033839828.svs 888.6 MB
1.070 GB

demo.kitware.com/histomicstk/...m/5bbdeed1e629140048d01bcb

TCGA
Crowd Source Paper
TCGA-OR-A5J8
TCGA-OR-A5J2
TCGA-OR-A5J1
TCGA-OR-A5J6
...

Collections: TCGA, Crowd Source Paper

Folders:
TCGA-OR-A5J7
TCGA-OR-A5J5
TCGA-OR-A5J4
TCGA-OR-A5J3
TCGA-OR-A5J2
TCGA-OR-A5J1
TCGA-OR-A5J6
TCGA-OR-A5J8
TCGA-OR-A5J9

Items:
TCGA-HT-7474-roi-nuclei.anot
TCGA-HT-7474-roi-nuclei-features.csv
TCGA-OR-A5J5-01A-01-TS1.338CEA5D-6428-4BFD-85CE-628FBC4180F8.svs
TCGA-OR-A5J1-01Z-00-DX5.145D832C-4AA8-43F4-A904-BC2C3B02073E.svs
TCGA-AN-A0G0-01Z-00-DX1.BE0BB5DF-DEDA-48D8-B5D8-2735C767F28F.svs
TCGA-OR-A5J1-01A-01-TS1.CFE08710-54B8-45B0-86AE-500D6E36D8A5.svs
TCGA-OR-A5J7-01Z-00-DX1.D8D52E9A-12F2-4FF7-9203-314E7AA7608F.svs
TCGA-OR-A5J5-01Z-00-DX1.4ED45C49-BD89-4DF0-BF92-37E79A3D244C.svs
TCGA-OR-A5J6-01Z-00-DX1.C3F415F4-B679-433F-B8C2-33ED940272FB.svs
TCGA-OR-A5J9-01A-01-TSA.13AE6172-F829-414C-B723-97A2E9CD2304.svs

Some screen-grabs from my attempt at navigating DSA.

The key takeaways:

- The *search* process is convoluted and complex. You can sort by Collection, Image, Metadata, or more. You can only select one of these options at a time — not aggregate multiple. You can use a special syntax for power searches.
- You can also group and organize images by *folder.* This occurs in a normal file-structure way: you can nest folders within others, leading to a "tree-like" traversal use pattern comprised of complex paths and nested relationships.

These are, critically, the only two ways to find images (!).

User Flows

To double check my work, I interviewed three people to gather some user flow data:

i. My manager, serving the role of an experienced user with years of using DSA.
ii. Jacob Siboyeh (Cornell '24), a co-worker on my team in an unrelated field (Computer Science) and thus no domain knowledge — thus serving the role of an inexperienced and unsystematic newbie to the platform.
iii. Ryan Ma (Berkeley '25), another co-worker with domain knowledge — serving the role of the inexperienced but systematic newbie.

The user flows looked something like this:

Manager Flow:

i. He instinctively reached for the folders first. His observation was that “if you set up the viewer right, folders are the fastest way to find an image.”

ii. Through experience, he’s able to navigate through the folders quite quickly. He for instance is able to quickly find specific images. When he “misses” a folder, either by misclick or by choosing the wrong folder, he is able to “backtrack” quickly (recognizing his error early on and navigating back up a level without wasting too much time diving deeper).

iii. He reaches for the search box as a last resort. This part is more difficult for him: he attempts to recall specific image names and often fails.

Jacob’s Flow:

i. Jacob, by contrast, reaches for the search box first, especially when trying to pinpoint a very specific given image.

ii. Given the image name, he finds it quickly. When not given the image name but only properties, he will attempt to search by property. It takes him a medium amount of time to reach the radio options that allows one to select what property to search by.

iii. Failing search, he will fall back to folder traversal. His traversal times are slower than Manager’s, and failed searches cost more in terms of time wasted and frustration.

Ryan’s Flow

i. Ryan, surprisingly, reaches for folders first. He takes his time to try and understand the given folder structure. It appears he is trying to get an intuitive sense for what might be where.

ii. This improves his traversal times when finding a specific image in the folder structure later, as he makes “folder misses” less often.

iii. He performs about the same as Manager on searches, but is able to recall memorized image names more quickly (this may be an arbitrary difference).

Based on conversations with my manager and clients, I also came across one more critical point:

- Users frequently create a folder named after the date that images were uploaded together, or after the date they anticipate using all of these images together, for quick access in batches in the future.

Pain Points

This leads to the following observations:

- This structure encourages a user behavior, where users group images using naming syntaxes for search convenience. Related images can have similar names that can then be searched for together using the special syntax they have enabled. A similar fact goes for structured metadata, etc. Think Dropbox Enterprise. This comes, critically, at the cost of human readability, and requires memorization (not that doctors are bad at it).
- For folder organization, the *upfront cost of poor organization is very high,* since a badly designed folder structure makes it unlikely to find an image in reasonable time in the future. Users *must make the tradeoff, sacrificing immense time right now, as the image uploads, in favor of less time in the future,* like a busy college student spending half his Sunday meal-prepping the rest of the week (me). Done well, there’s no choice paralysis at any folder level at any level — the user simply makes the optimal choice all the way “down” through the folder layers to find their image. Done poorly, and the backtracing/scouring is simply untenable.

Part 2: Redesign Goals

Measuring Success & KPIs

Let’s measure success by *the average time it takes to get to a specific image*. A specific image can either be given by name, or some combination of unique identifying properties (metadata, appearance, etc.) Our goal is to beat DSA — hopefully by a large margin. In the future, this metric can be automatically tracked using Google Analytics, PostHog, or some similar analytics service.

Hypotheses

Instead of a hierarchical, folder-based approach to image organization, which encourages high-cost misses, I will take a “horizontal” approach where all images are on the same level, but you can group them by *tag.* I believe this is the superior approach for a few reasons, including:

- With tags, it only takes ~1 click to get to an image group, rather than 2+ for folder navigations.
- You can assign an image to multiple groups, rather than forcing it to only exist in one folder (or duplicating an image across multiple folders manually).
- There is almost zero cost to a wrong search (just try another tag).
- You can search multiple groups at once (apply multiple tags in your search).

After getting this decision approved by my manager, I proceeded to address the remaining problems:

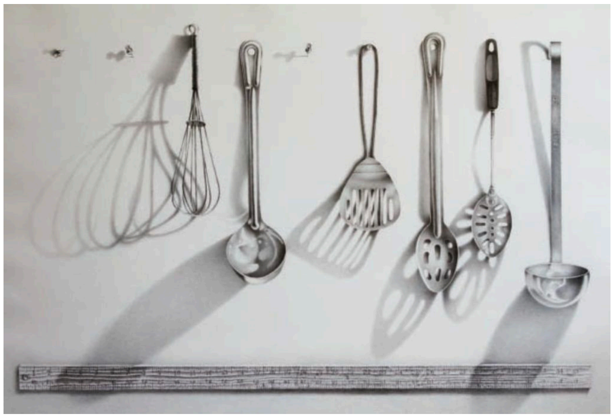
- Search will automatically apply to an image’s name, but not metadata (Instapath users are discouraged from touching raw metadata; existing users rarely touch it already). To adapt for those that were using metadata to store non-name information, I added an optional Description field that is also searchable by default.
- I added a new filter: you can now filter your search results by Date Range uploaded— leaning into the natural chronological-proximal nature of user behavior.
- All filters are combinable: search + date filter + tags.
- If the combination is still not specific enough (too generalized, as opposed to the manual atomic specificity of folders), one can now sort by name, upload time, etc., within the search results, to bring relevant results closer to the top of the page.

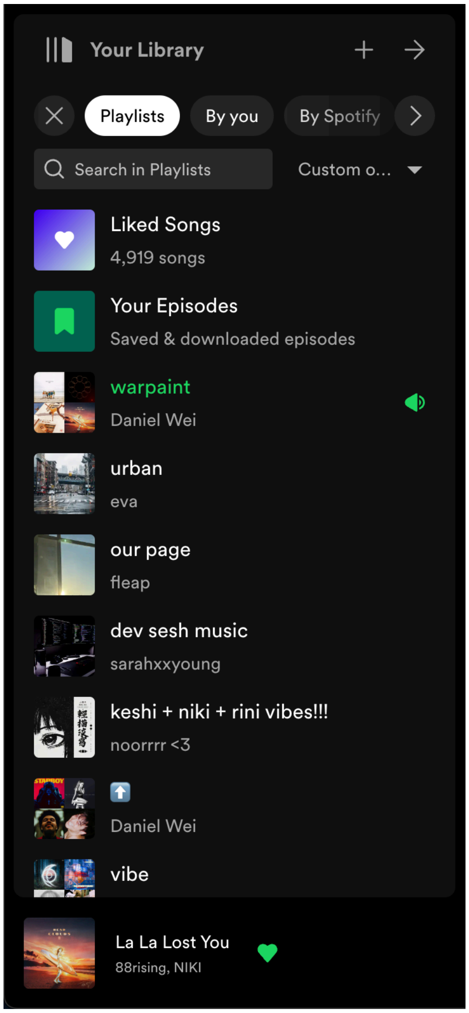
This combination of features should allow for a new and improved user flow.

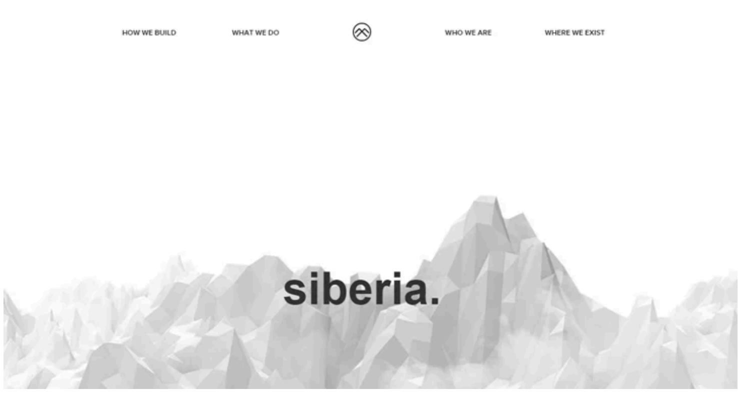
Part 3: Brainstorming

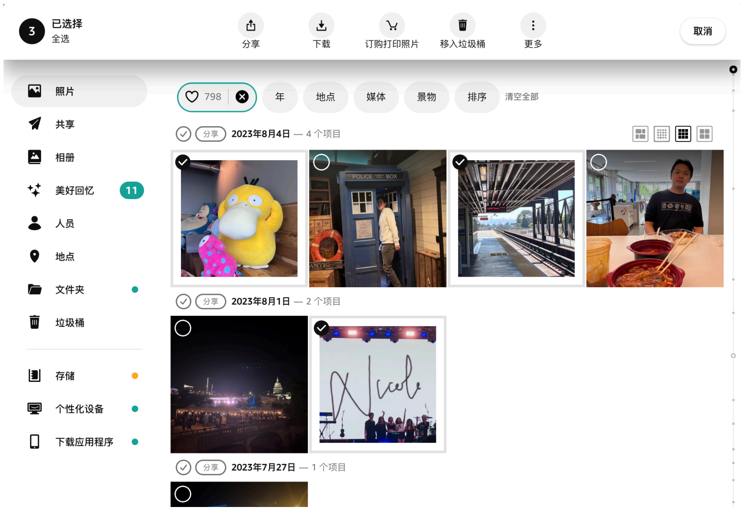
Speculative Design

Reminiscent of Dark Rooms in photography, or the painting equivalent of a Graphite Still-Life.
Design precedents: Spotify and Amazon Photos. Similar to: Soft minimalist web portfolios.
Vibes: round, soft, malleable, charcoal, pills, medicine, clean-washed.





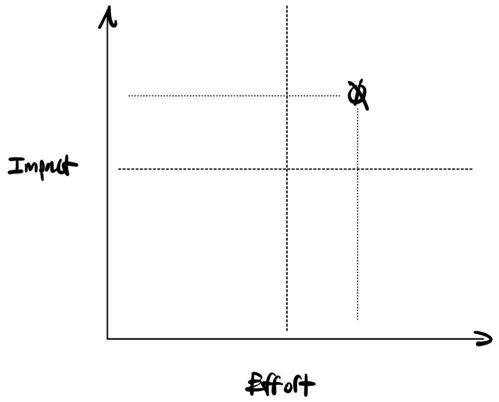




Impact-Effort Matrix

As a program, I'd rate the difficulty of implementation at medium-high. Implementing a feature like this cleanly involves a core architecture rewrite using what we call a "reducer" — a way to calculate future search results based on a mathematical-ish combination of user decisions.

However, I'd also rate the impact as medium-high. Perhaps the only experience less critical than viewing an image, is the process by which you get to the image in the first pace. This is backed up by observation data.

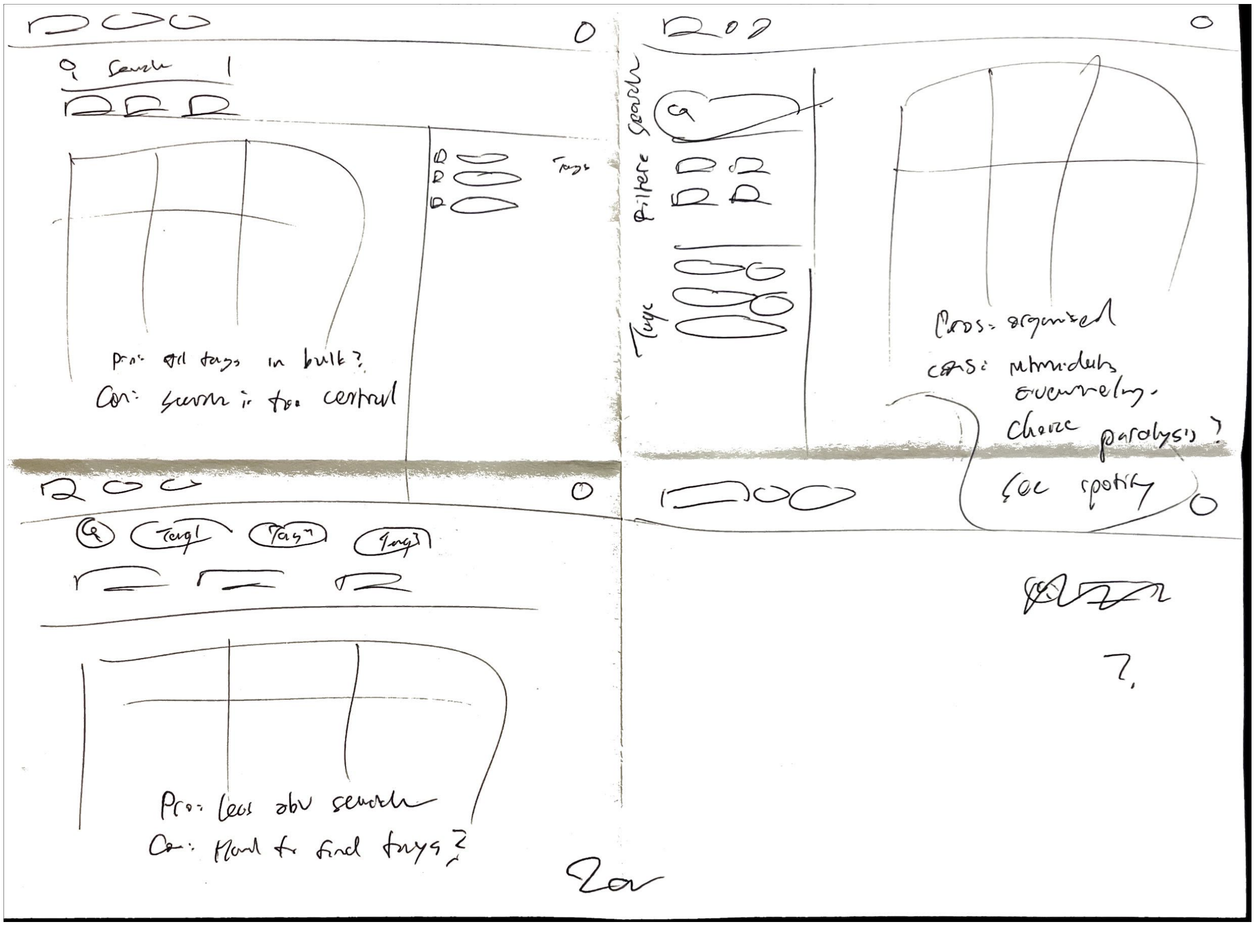


Lo-fi Explorations

I used to be an art student, so I prefer pencil and paper. Unfortunately, this combined in bad ways with the STEM inclination to write as quickly (read: messily) as possible, so apologies in advance.

Notes: I usually do my explorations on an A4 paper folded in quadrants. I aim to put one exploration in each quadrant. Here, I explore, starting with the top right quadrant being quadrant I and continuing counter-clockwise: in Quadrant I: a collapsible sidebar with three distinct sections for search, tags, and filters; in Quadrant II: search and filters along the top, tags along the right; in Quadrant III: everything in one top bar, with a row for tags and a row for search and filter. Some pros and cons are listed for each. They can be summarized as: I is intimidating and a poor use of screen real estate for small computers; II is chaotic (arbitrary division), favors search too much (user flows showed a preference for innate groupings), and is an even worse use of screen real estate; III is perfect for screen real estate and aptly deprioritizes search, but it may be hard to find tags in the future if there are too many (horizontal scrolling).

An additional note: for the Lo-fi Exploration in Quadrant III, there's some influence from how Spotify designed their new desktop Library experience. I think the fundamental takeaway from how controversial it was upon release is that most people do not have particularly large music libraries — it was an experience designed to unify the experience across mobile and desktop for *music hoarders,* not the vast majority of users that have ~5-10 playlists at most. As someone that has hundreds of playlists and follows >1000 artists, the new UI is fairly effective at filtering for what I'm looking for.



Pro: all tags in bulk?
Con: search is too central

Pros: organized
Cons: information overwhelming.
Choice paralysis?
See spotify

Pro: less abt search
Con: Hard to find tags?

Part 4: Finalization

User Feedback

Working under a compressed timeline, I did not have time to build out MVPs for everything. Instead, since flows I and II are fairly similar (search prioritization, vertical tags), I only built out an MVP for one of them (Flow II). I then also built Flow III, and ran extensive user testing on both (internal usage by the entire Instapath team — I watched my manager run it; the rest of the team took notes and forwarded them to me for compressed-timeline reasons).

User feedback and observations summary for Flow II:

- Exceedingly easy to search.
- Found the date range selector exceedingly useful (latent need addressed!).
- With lots of tags, requires some vertical scrolling to reach a tag, select it, and then scroll back up to see what image results came up (!). *This was something I had not anticipated. We could work around it with an extra search bar just for tags, but that seems potentially confusing for users (why would we have two search bars ?).*

User feedback and observations summary for Flow III:

- Lots of horizontal scrolling to find tags (anticipated) but prioritization of tags over search (in fact, such high prioritization that sometimes they'd select tags before the images had even finished loading in — a purely technical challenge).

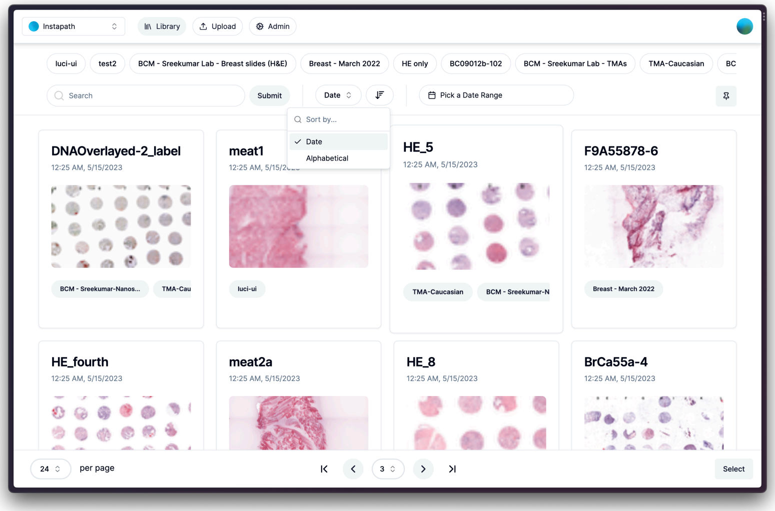
- Found the date range selector exceedingly useful as well.
- Likely to use a combination of factors to filter out images more specifically (possibly because options are all grouped in the same area on the page). *This is good!*

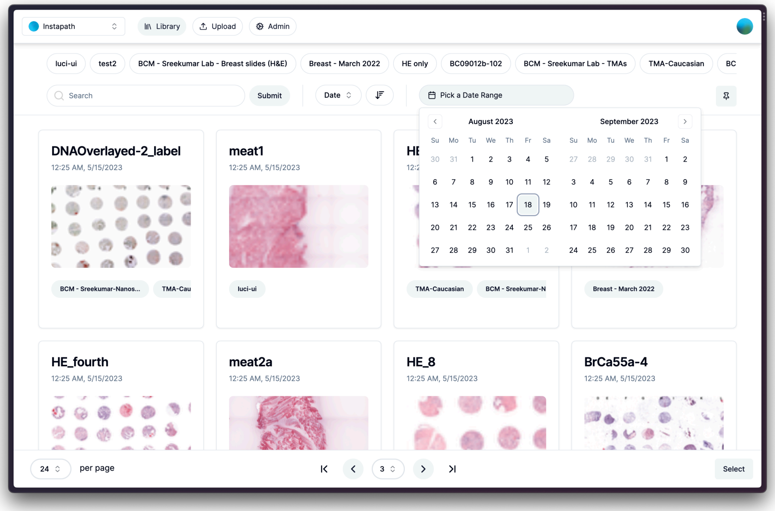
Selecting a Flow

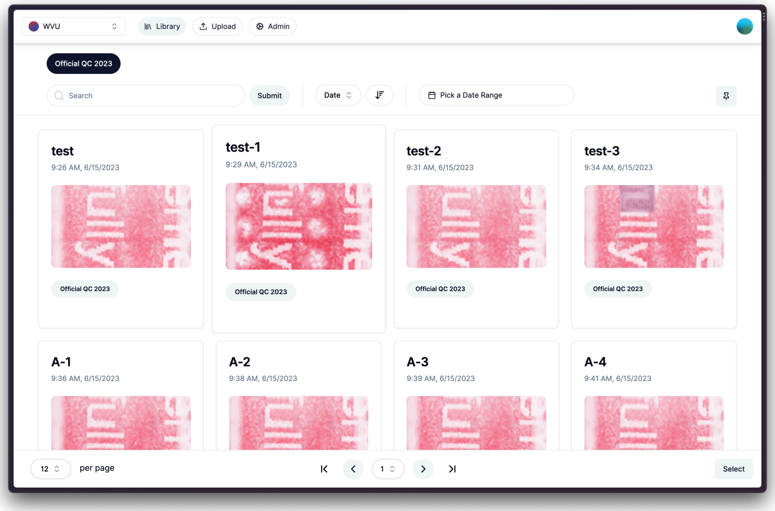
I elected to go with Flow III for a few reasons:

- Users wasted time scrolling down and up for Flow II, which was only slightly more cumbersome than the excessive horizontal scrolling of Flow III. Searching for tags? Out of the question.
- Likely to use factors in combination.

Hi-Fi Result







test
9:26 AM, 6/15/2023
Official QC 2023

test-1
9:29 AM, 6/15/2023
Official QC 2023

test-2
9:31 AM, 6/15/2023
Official QC 2023

test-3
9:34 AM, 6/15/2023
Official QC 2023

A-1
9:36 AM, 6/15/2023

A-2
9:38 AM, 6/15/2023

A-3
9:39 AM, 6/15/2023

A-4
9:41 AM, 6/15/2023

Part 5: Conclusion

Unsolved Problems

- For organizations, both horizontal scrolling (as designed and implemented in the final High Fidelity) and vertical scrolling to access tags is a problematic inconvenience. I anticipate adding a special search box for tags to be excessive and confusing. Perhaps a multi-combobox dropdown, with the option to select and search for multiple tags upon entering the combobox, might suffice, but it would require further research, as that's a pretty non-standard UI tool typically reserved for space constrained input forms (i.e. job application "Skills" inputs).

Case Study 2: Image Switching

For many of the features available in the final version of the Luciviewer that we ended up designing and building, the design was fairly intuitive. After all, when in doubt, play it safe by sticking to tried-and-true design patterns. Thus, search pagination looks like a regular paginator; user and global settings tables look like every other table out there; the log out button is the same one used on half the world wide web.

A more serious problem comes up with what appears at first to be a simple issue: how do we enable switching quickly between multiple, related images, without having to go back to your Search Results first and then clicking on another image (>5s per cycle)? For instance, a doctor may want to quickly cycle through all the scans for the patients he will be seeing within the next hour; a nurse may want to pull up all images of one certain rare disease together. Enabling this behavior is not as intuitive as it may seem; Open in New Tab isn't exactly good UX.

Parts

Part 1: Research

Competitor Analysis

As in the first case study, our only competitor in this field is the industry standard Digital Slide Archive (DSA) project.

Now, as it turns out, DSA simply has no implementation of quick switching. The closest thing it has is perhaps folder views — if you take the time to arrange your photos in an organized manner before hand, grouping similar images together, you can more quickly switch between images in the same folder.

User Flows

From light experimentation with colleagues in the workplace, I found the following simple user-flows in DSA:

- Many people would right-click-Open-in-New-Tab all relevant images.
- Some people would take the time to organize relevant images into the same folder and thus make switching between those images slightly faster.

After implementing an MVP of the new interface (see Case Study 1) *without* any image-switcher, the *only* user flow I could observe among all clients and teams that used the product was this:

- Users would click on Library, or the Organization Name, to go back to All Images. After waiting for the page to load and then all images to load, they would repeat their previous search and then attempt to find the next image they wanted to navigate to.

Pain Points

As is obvious, the main pain point is the time it takes to switch between images that the doctor finds related, as well as the complete context loss/switch that occurs each time that happens (for instance, if after navigating from A to B, it's difficult to navigate back to A, that's also a design failure).

Part 2: Redesign Goals

Measuring Success

It's fairly evident that our goal is to make it easy to navigate from Image A to related Image B, where relation is defined by the needs of the doctor (eg. Images that are of the same patient, uploaded within a similar time frame, have similar names, are of the same disease, or etc.). However, that's not enough — it must also be easy to navigate from Image B *back* to Image A, in the same way that one might Alt+Tab quickly between two images. Let's define success metrics, or our KPI, then, as the amount of time it takes to go from a specific image to another, and then back to the former.

Hypotheses

- To maximize FOV, whatever image-switching interface we use has to be hidden by default, and only pop out when needed.
- Users often want to go back to their search results and then choose an image adjacent within the search results to the one previously selected.
- Their *latent need*, however, is to find a set of closely related images, that they can switch between. Although limited by the upper end of what a technical programmer can implement algorithmically, we can make use of the power of CS here to serve relevant images as well, based on the current image and other factors.

This boils down to the following UX hypotheses:

- A slide-in slide-out panel with relevant images can be implemented, trigger either by a simple on-screen button (to maximize FOV) or a keyboard gesture (a more habit-forming, power-user-friendly gesture similar to Alt+Tab).
- The panel should distinguish between *manually added images,* which the user can manually claim are related to the image they're currently on and should be able to easily access; and *suggested images,* which are determined by us, and are comprised of search results (naturally closely related to the current image) and perhaps some algorithmically driven

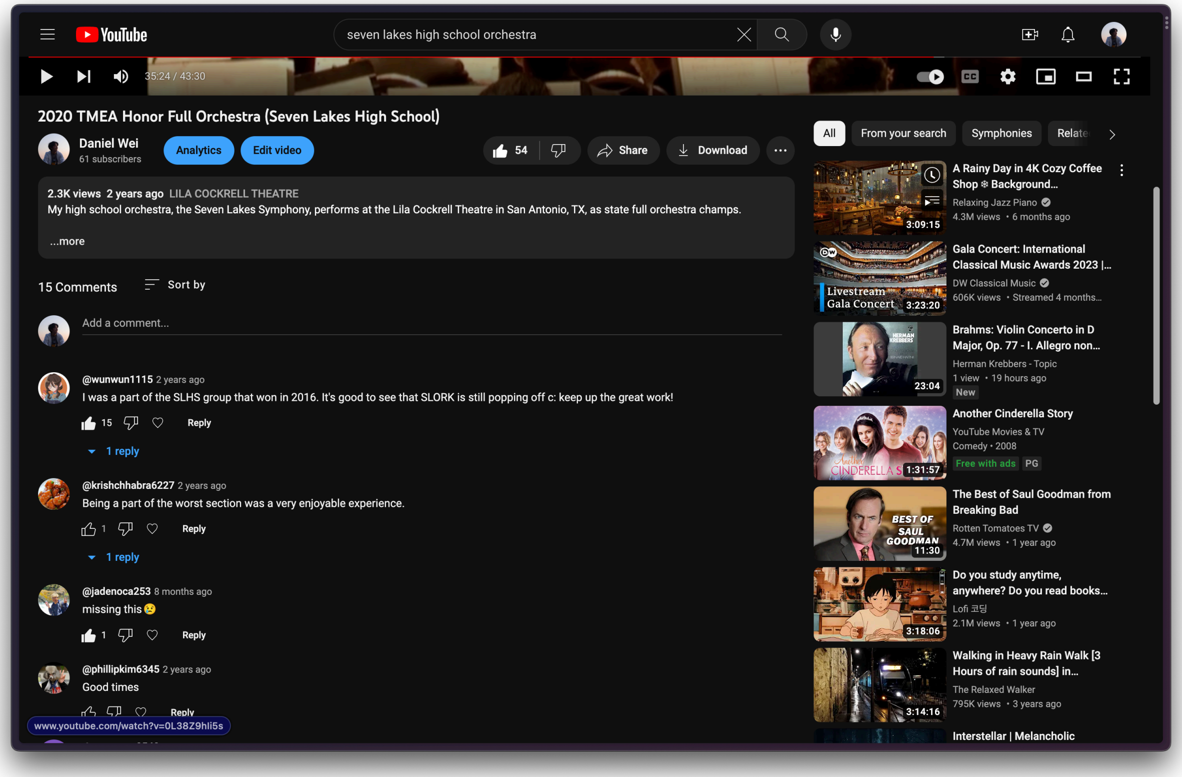
content (to anticipate user needs and cut down time spent searching through search results).

Part 3: Brainstorming

Design Precedents

To generate some ideas for this idea, I specifically looked to see how YouTube implements a similar feature. The reasons for this are:

- YouTube has two modes: video mode (YouTube) and music mode (YT Music). The two are essentially the same, except for substantial UI differences.
- In YouTube, by default, when you view a video, only recommended videos are suggested alongside the current video, based on the current video.
- However, in YouTube Music, when viewing a song, you by default play the next song in the playlist (manually created), queue (manually created), or search results (manually queried).
- Critically, both are only default behaviors, and with some extra work, support the additional functionality of the other platform. You *can* create playlists and queue up videos in YouTube, it's just a little more hidden away than in YT Music. Similarly, you *can* find autogenerated recommendations similar to the current song in YT Music — it's just more hidden away than in YouTube.
- Our goal is to achieve a balance inspired by the designs of YouTube and YouTube Music, but which makes it possible to easily switch to manually-determined or automatically-determined relevant images!



2020 TMEA Honor Full Orchestra (Seven Lakes High School)

Daniel Wei
61 subscribers

2.3K views 2 years ago LILA COCKRELL THEATRE
My high school orchestra, the Seven Lakes Symphony, performs at the Lila Cockrell Theatre in San Antonio, TX, as state full orchestra champs.
...more

@wunwun1115 2 years ago
I was a part of the SLHS group that won in 2016. It's good to see that SLORK is still popping off c: keep up the great work!

@krishchhabra6227 2 years ago
Being a part of the worst section was a very enjoyable experience.

@jadenoca253 8 months ago
missing this 😢

@phillipkim6345 2 years ago
Good times

A Rainy Day in 4K Cozy Coffee Shop ❄ Background...
Relaxing Jazz Piano
4.3M views • 6 months ago
3:09:15

Gala Concert: International Classical Music Awards 2023 |...
DW Classical Music
606K views • Streamed 4 months...
Livestream Gala Concert 3:23:20

Brahms: Violin Concerto in D Major, Op. 77 - I. Allegro non...
Herman Krebbers - Topic
1 view • 19 hours ago
New
23:04

Another Cinderella Story
YouTube Movies & TV
Comedy • 2008
Free with ads PG
1:31:57

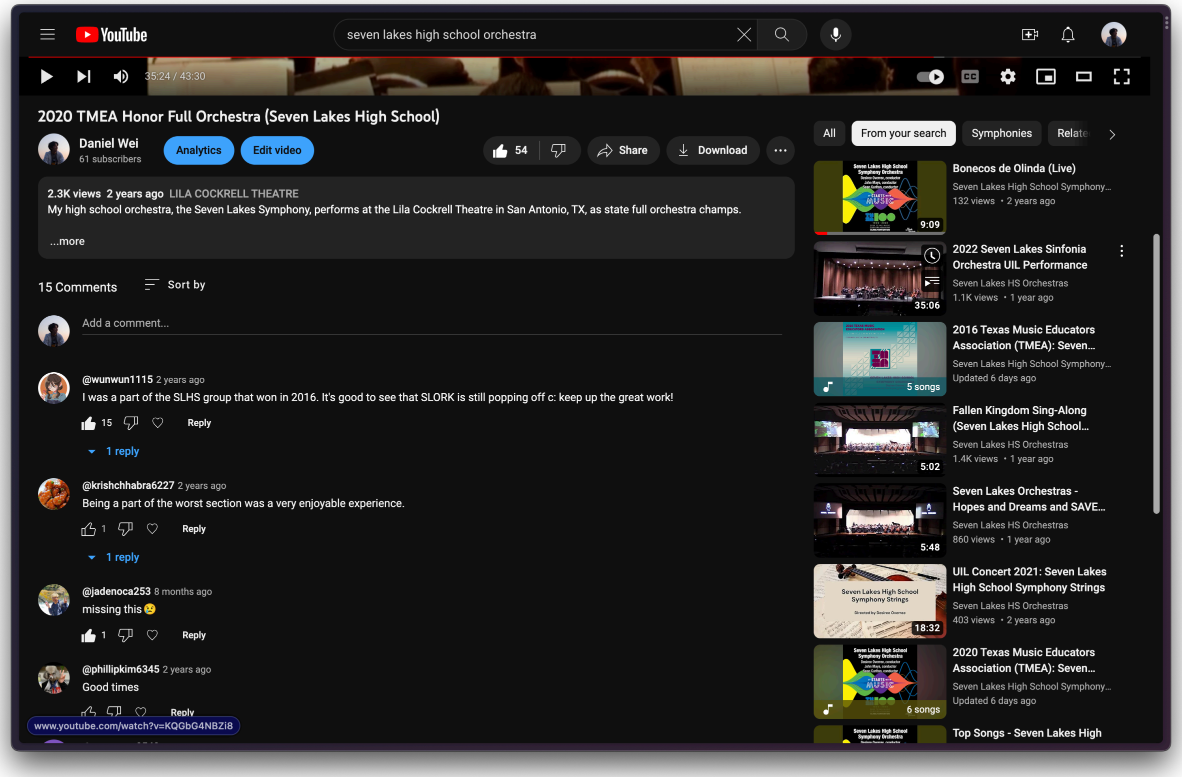
The Best of Saul Goodman from Breaking Bad
Rotten Tomatoes TV
4.7M views • 1 year ago
BEST OF SAUL GOODMAN 11:30

Do you study anytime, anywhere? Do you read books...
Lofi 코딩
2.1M views • 1 year ago
3:18:06

Walking in Heavy Rain Walk [3 Hours of rain sounds] in...
The Relaxed Walker
795K views • 3 years ago
3:14:16

Interstellar | Melancholic

Note how by default, a list of automatically suggested recommended videos are suggested beside any video.



2020 TMEA Honor Full Orchestra (Seven Lakes High School)

Daniel Wei
61 subscribers

2.3K views 2 years ago LILA COCKRELL THEATRE
My high school orchestra, the Seven Lakes Symphony, performs at the Lila Cockrell Theatre in San Antonio, TX, as state full orchestra champs.
...more

@wunwun1115 2 years ago
I was a part of the SLHS group that won in 2016. It's good to see that SLORK is still popping off c: keep up the great work!
15

@krishchhabra6227 2 years ago
Being a part of the worst section was a very enjoyable experience.
1

@jadenoca253 8 months ago
missing this 😢
1

@phillipkim6345 2 years ago
Good times

Bonecos de Olinda (Live)
Seven Lakes High School Symphony...
132 views • 2 years ago
9:09

2022 Seven Lakes Sinfonia Orchestra UIL Performance
Seven Lakes HS Orchestras
1.1K views • 1 year ago
35:06

2016 Texas Music Educators Association (TMEA): Seven...
Seven Lakes High School Symphony...
Updated 6 days ago
5 songs

Fallen Kingdom Sing-Along (Seven Lakes High School...
Seven Lakes HS Orchestras
1.4K views • 1 year ago
5:02

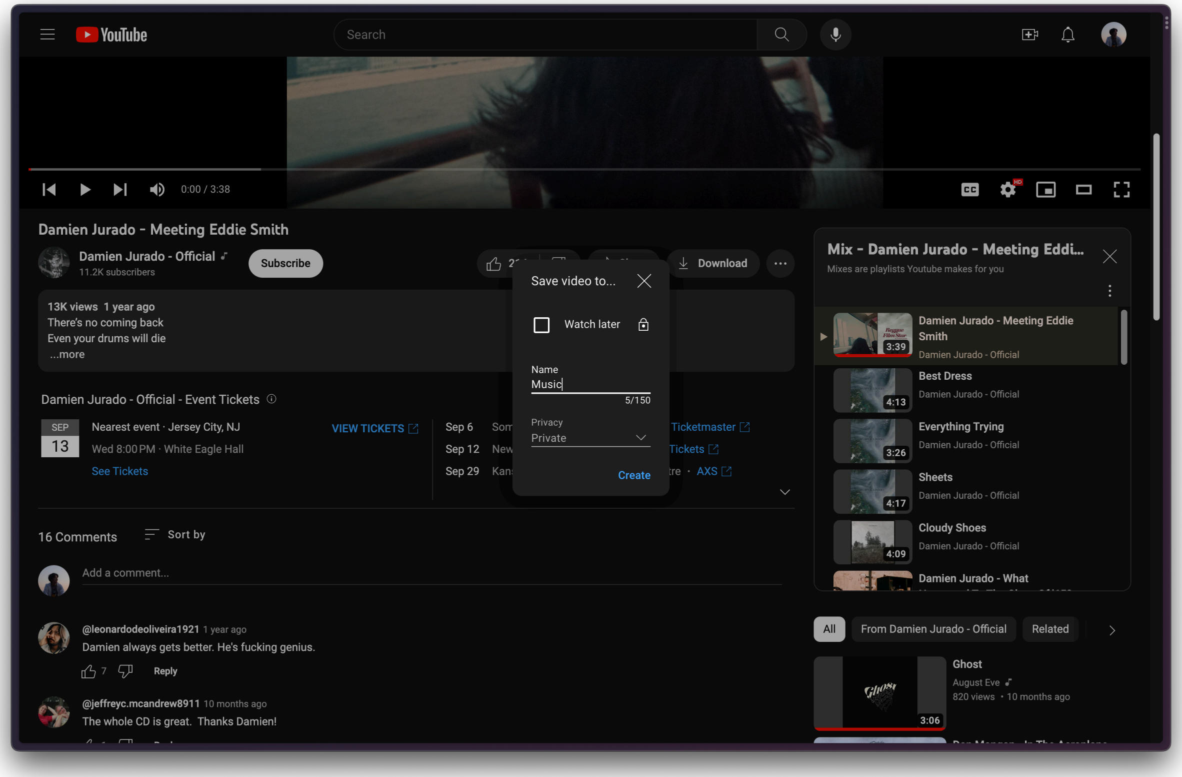
Seven Lakes Orchestras - Hopes and Dreams and SAVE...
Seven Lakes HS Orchestras
860 views • 1 year ago
5:48

UIL Concert 2021: Seven Lakes High School Symphony Strings
Seven Lakes HS Orchestras
403 views • 2 years ago
18:32

2020 Texas Music Educators Association (TMEA): Seven...
Seven Lakes High School Symphony...
Updated 6 days ago
6 songs

Top Songs - Seven Lakes High

Additional options let you swap out this generated list for the list of search results that you got on the previous page with one click, reducing the time it takes to go back and forth between search results.



Damien Jurado - Meeting Eddie Smith

Damien Jurado - Official
11.2K subscribers

13K views 1 year ago
There's no coming back
Even your drums will die
...more

Damien Jurado - Official - Event Tickets

SEP 13 Nearest event · Jersey City, NJ
Wed 8:00 PM · White Eagle Hall

Sep 6
Sep 12
Sep 29

@leonardodeoliveira1921 1 year ago
Damien always gets better. He's fucking genius.
7

@jeffreyc.mcandrew8911 10 months ago
The whole CD is great. Thanks Damien!

Mix - Damien Jurado - Meeting Eddi...
Mixes are playlists Youtube makes for you

Damien Jurado - Meeting Eddie Smith
Damien Jurado - Official
3:39

Best Dress
Damien Jurado - Official
4:13

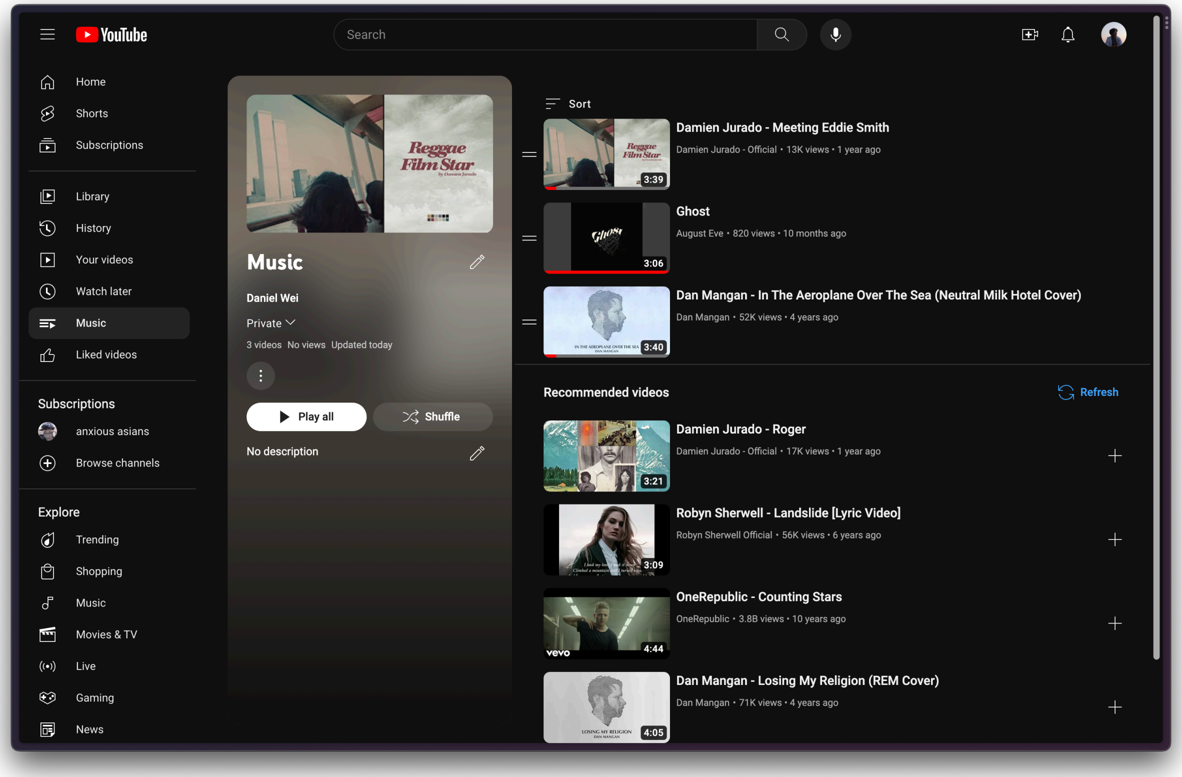
Everything Trying
Damien Jurado - Official
3:26

Sheets
Damien Jurado - Official
4:17

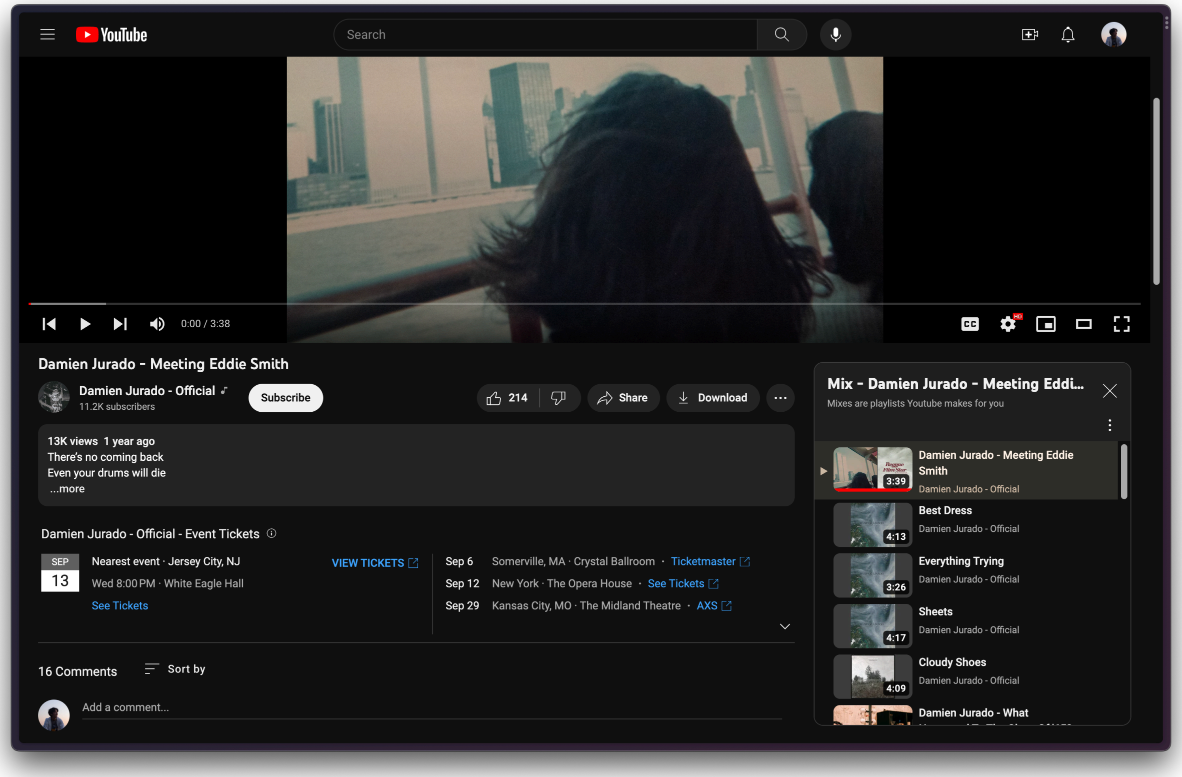
Cloudy Shoes
Damien Jurado - Official
4:09

Damien Jurado - What

Ghost
August Eve
820 views • 10 months ago
3:06

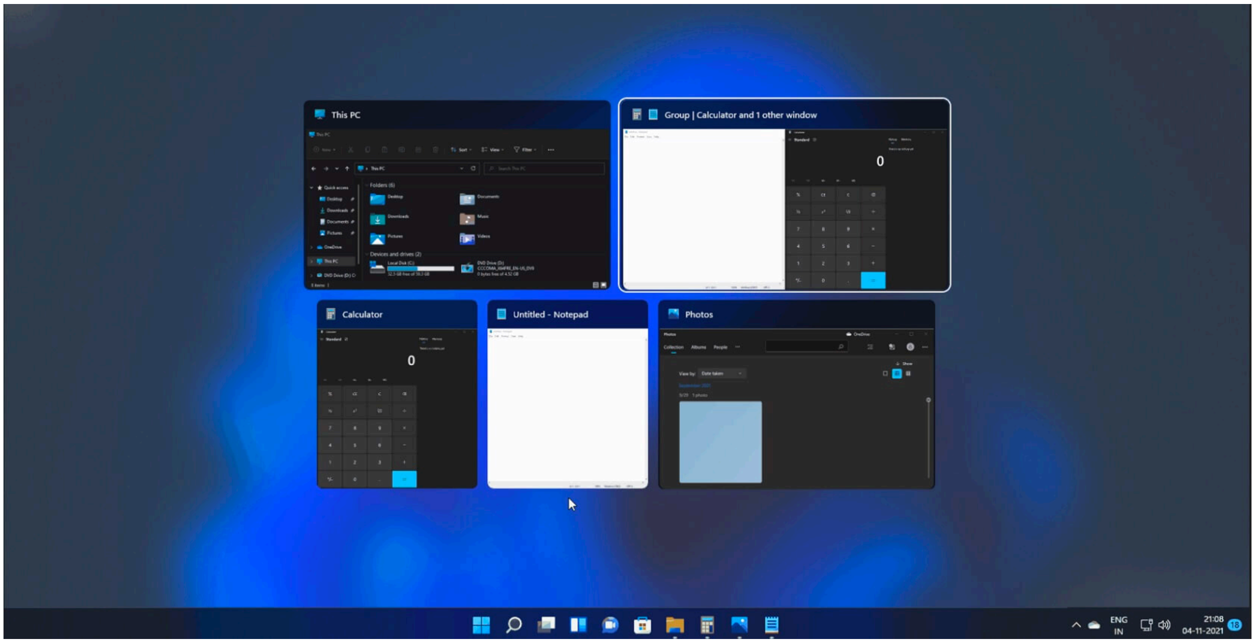

YouTube
Search
Home
Shorts
Subscriptions
Library
History
Your videos
Watch later
Music
Liked videos
Subscriptions
anxious asians
Browse channels
Explore
Trending
Shopping
Music
Movies & TV
Live
Gaming
News
Music
Daniel Wei
Private
3 videos No views Updated today
Play all
Shuffle
No description
Sort
Damien Jurado - Meeting Eddie Smith
Damien Jurado - Official • 13K views • 1 year ago
3:39
Ghost
August Eve • 820 views • 10 months ago
3:06
Dan Mangan - In The Aeroplane Over The Sea (Neutral Milk Hotel Cover)
Dan Mangan • 52K views • 4 years ago
3:40
Recommended videos
Refresh
Damien Jurado - Roger
Damien Jurado - Official • 17K views • 1 year ago
3:21
Robyn Sherwell - Landslide [Lyric Video]
Robyn Sherwell Official • 56K views • 6 years ago
3:09
OneRepublic - Counting Stars
OneRepublic • 3.8B views • 10 years ago
4:44
Dan Mangan - Losing My Religion (REM Cover)
Dan Mangan • 71K views • 4 years ago
4:05

Alternatively, you can create and manage a playlist of user-determined videos.

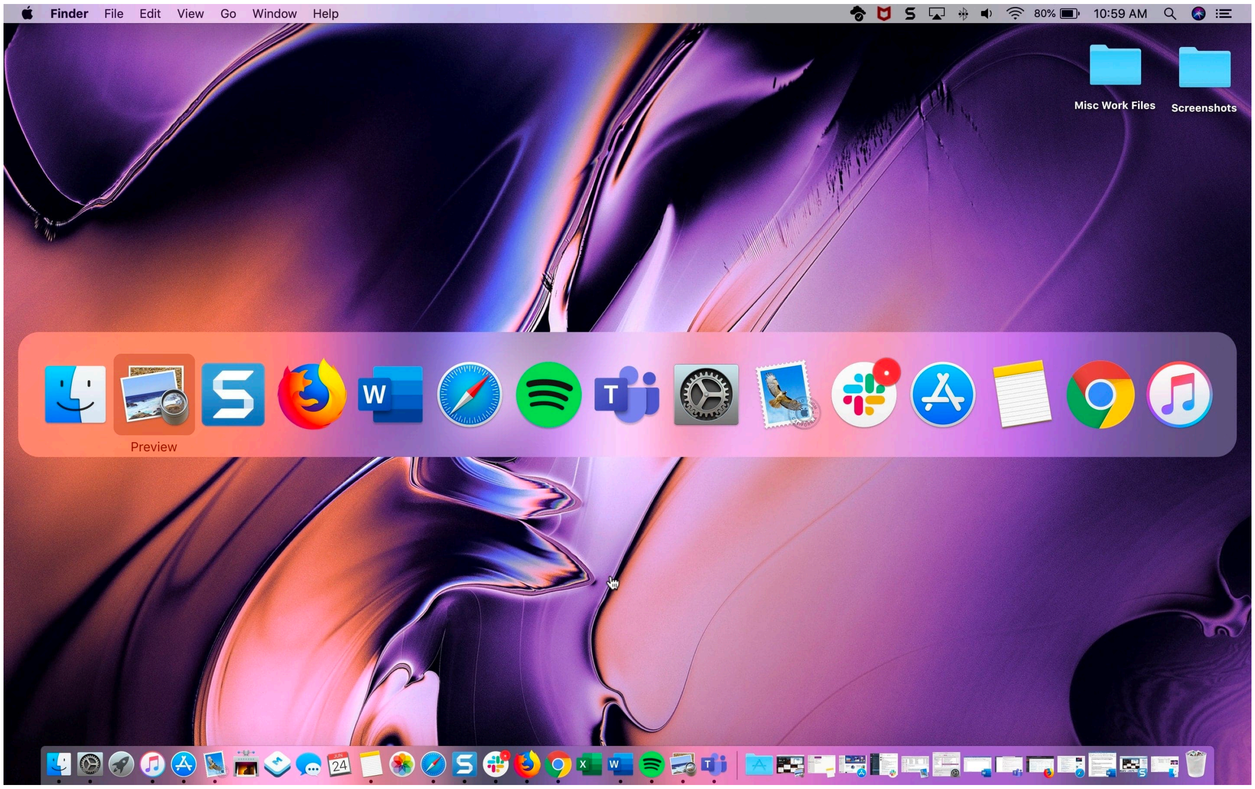


Songs can then be added to a queue, which appears as a box in the bottom right, on top of where recommended images usually lie.

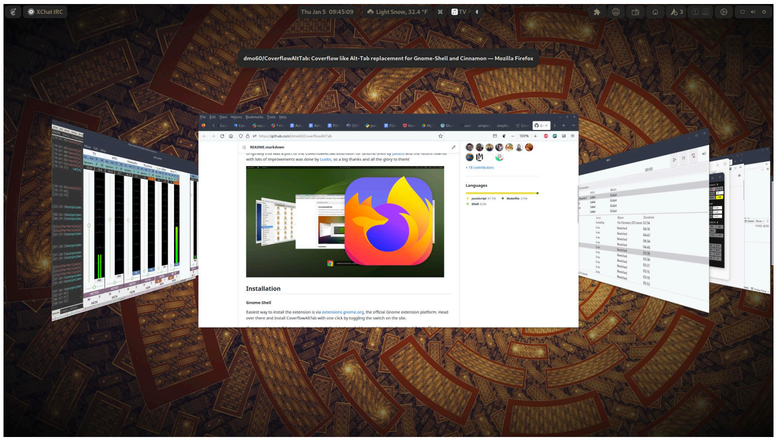
Some more inspiration can be taken from how different operating systems handle Alt+Tab (or Cmd+Tab) — a way to quickly switch between windows without losing context is a problem that shares many features with ours.



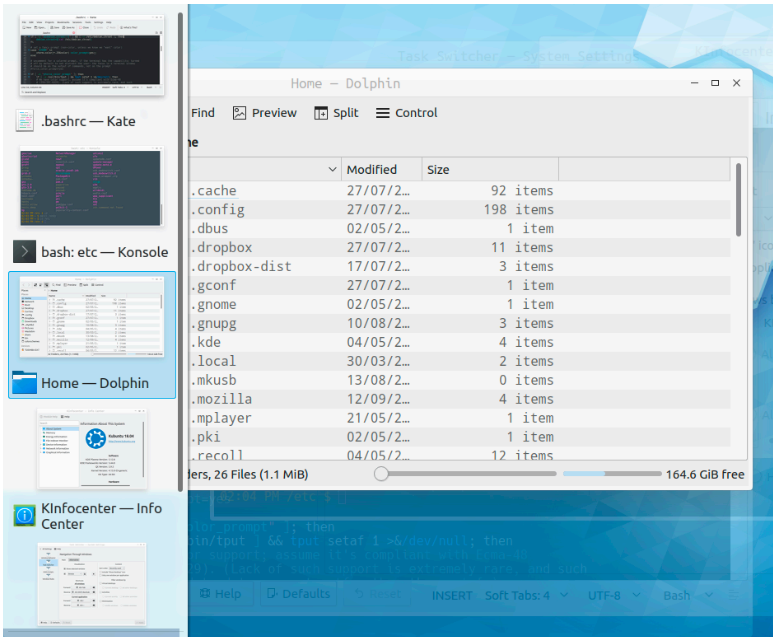
Here, Windows 11 has a non-deterministic grid of windows you can tab through, each with a name and preview.



By contrast, Mac OSX displays only icons, reducing the ability to preserve context and tell at a glance what was going on in each application. However, they implement a more linear/cyclic UI for tabbing through.


XChat IRC
Thu Jan 5 09:45:09
Light Snow, 32.4 °F
TV
dmo60/CoverflowAltTab: Coverflow like Alt-Tab replacement for Gnome-Shell and Cinnamon — Mozilla Firefox
README.markdown
Installation

Some Linux desktops feature a full width window-switcher that cycles through previews of windows. However, it's difficult to tell at a glance just how many windows are open on this desktop in total.



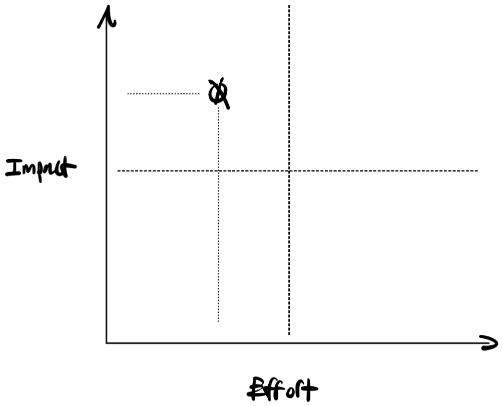
	Modified	Size
.cache	27/07/2...	92 items
.config	27/07/2...	198 items
.dbus	02/05/2...	1 item
.dropbox	27/07/2...	11 items
.dropbox-dist	17/07/2...	3 items
.gconf	27/07/2...	1 item
.gnome	02/05/2...	1 item
.gnupg	10/08/2...	3 items
.kde	04/05/2...	4 items
.local	30/03/2...	2 items
.mkusb	13/08/2...	0 items
.mozilla	12/09/2...	4 items
.mplayer	21/05/2...	1 item
.pki	02/05/2...	1 item
.recoll	04/05/2...	12 items

ders, 26 Files (1.1 MiB) 164.6 GiB free

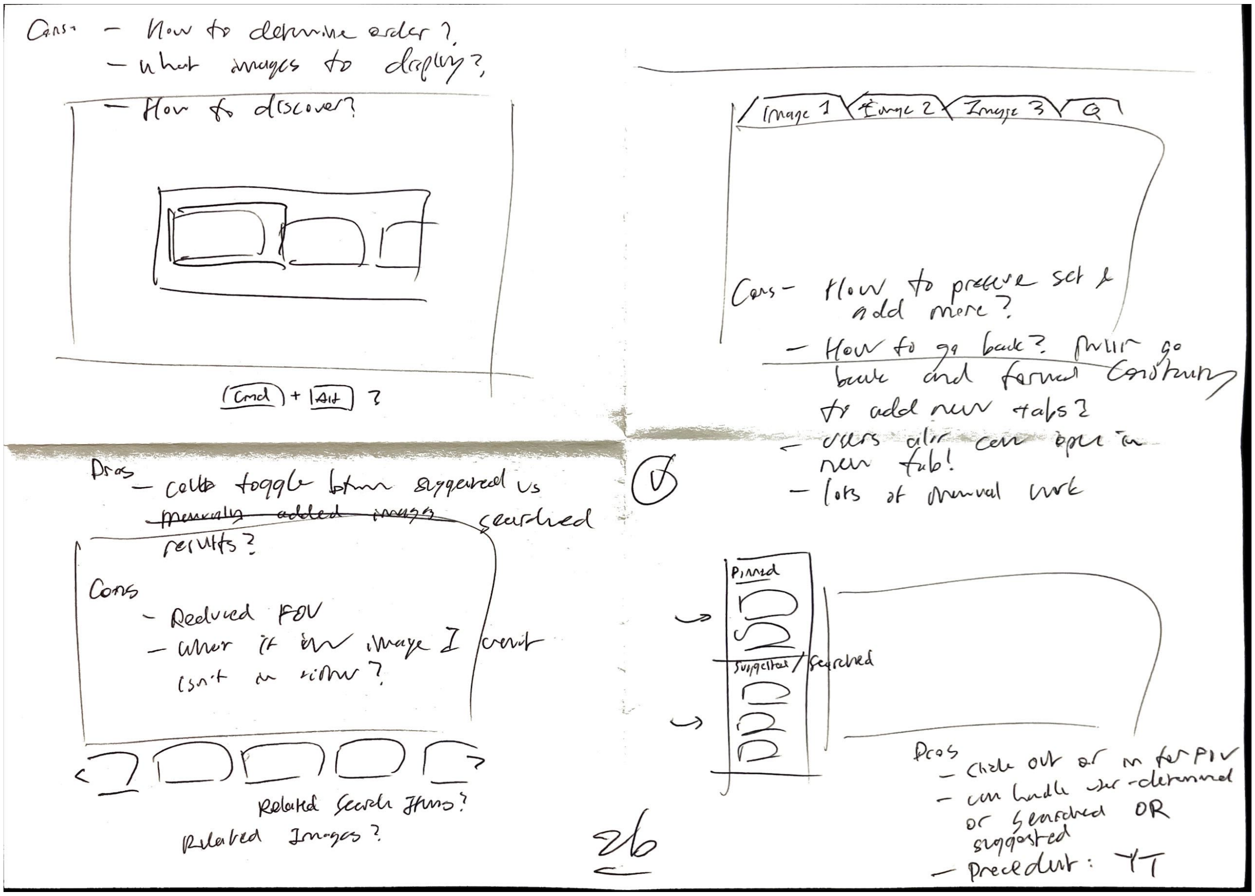
A simpler option is the so-called "KDE" Linux skin, which features a simple vertical, deterministic, linear window switcher, with the icon and name for each window below the window. Not the prettiest, but functional and predictable for the end user, with all the information you need at your fingertips.

Impact vs. Effort Matrix

This one's simple! High impact, low effort. My programmer side says he could crank a feature like this out in a day, tops.



Low Fidelity Sketches


Cons - How to determine order?
- what images to display?
- How to discover?
Cmd + Alt ?
Pros - could toggle btwn suggested vs ~~manually added images~~ searched results?
Cons
- Reduced FOV
- what if the image I want isn't in view?
Related Search Items?
Related Images?
26
Image 1
Image 2
Image 3
Q
Cons - How to preserve set & add more?
- How to go back? ~~How to go back and forward continuing to add new tabs?~~
- users already can open in new tab!
- lots of manual work
Pinned
Suggested / Searched
Pros
- Click out or in for FOV
- can handle user-determined or searched OR suggested
- Precedent: YT

Top Left: I consider a pop-over dialog, which takes the foreground and covers other content. Triggered by a one-handed keyboard combo similar to Alt+Tab, with no other affordance to indicate its existence, it's hidden away and only meant to be displayed during the brief liminal stage between images.

Con: how would I determine what options are available to tab to? Tabbing through search results is one thing if you have only a few, but what if there are hundreds? Do you even know if the image you want is in there?

Pro: it's really easy to build up a list of images that are quickly navigated between! Since Alt+Tab-like behavior brings the last-used image to the front, over time, the set of images at the front of the interface are the most frequently used!

Top Right: I consider a tabbed interface, where you can manually enter, rearrange, and remove relevant images.

Con: this requires a lot of manual user effort to maintain and be useful.

Pro: power users might enjoy it! A tabbed interface is highly customizable, and as long as they can find every image they want to add as a tab within a reasonable time frame, creating a set of images shouldn't take too long, and then after doing the work, each image is just one click away. Additionally, discoverability isn't too bad. Maybe we can derive something from the Tabs?

Bottom Left: I consider a slideshow-like interface, with a central image and tab-like previews of related images along the bottom that you can switch to.

Con: What images do we show at the bottom? Search results? Recommended images? To make it clearer, we could tab the tabs, but that would be too convoluted. Additionally, the interface takes up quite a bit of screen real estate, reducing our FOV (Field of View) for the central image at hand.

Pro: Context loss is minimized with image previews, and it seems to combine the best of the tabbing and alt-tabbing interface.

Bottom Right: I consider a KDE-like sliding panel that takes the foreground when switching between images, but which a) doesn't cover the current image completely, thus minimizing another context switch, and b) is split into two sections, in a similar way to the YouTube Queue: a manually-determined section, and a suggested section, which has a much clearer horizontal-tab for Suggested vs. Search Results, and then an infinitely scrollable list of relevant images below, along with each of their previews, below it. Finally, it should be trigger-able both via a discreet button placed along the toolbar in a corner of the current image, as well as by a keyboard combination in a manner similar to Alt+Tab.

This mitigates all the problems and strikes a general balance, at the cost of complexity — the average number of clicks is now 2, not 1, and could rise to as much as three (in order to pin an image by hand), and requires the user to navigate two sections of equal importance in the information hierarchy — which could lead to further wasted time in decision paralysis. Nonetheless, we chose to move forward with this design, at least for the time being.

Part 4: Finalization

User Feedback

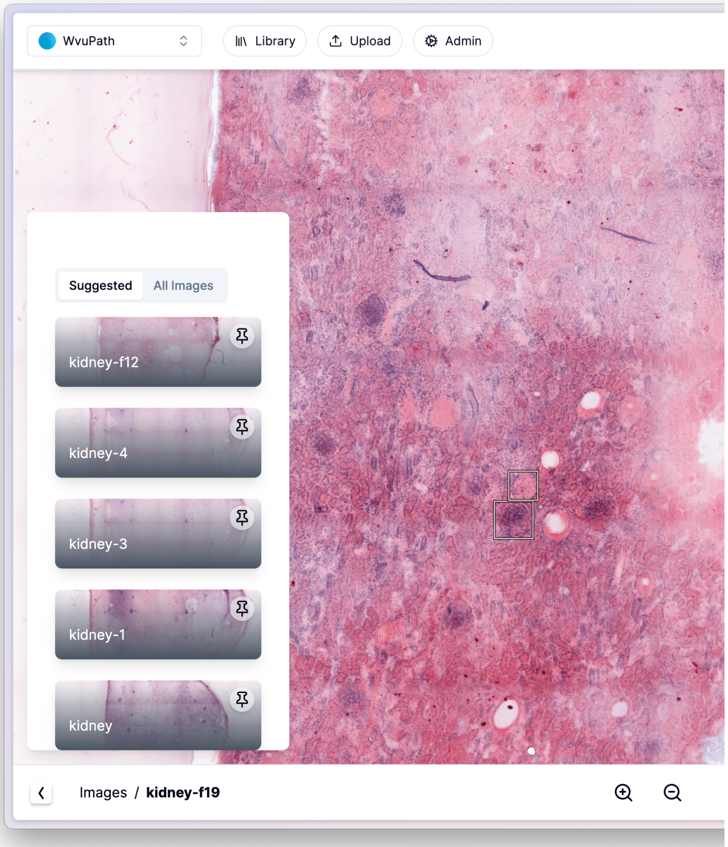
I tested a mock-up of this feature with three company members: my manager, the outreach manager that works directly with clients, and a regular hardware engineer. They have varying outlooks on software, experience with the product, and experience with our specific clients, making for a fairly diverse test group.

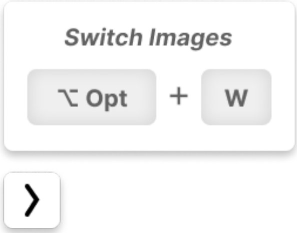
Observations:

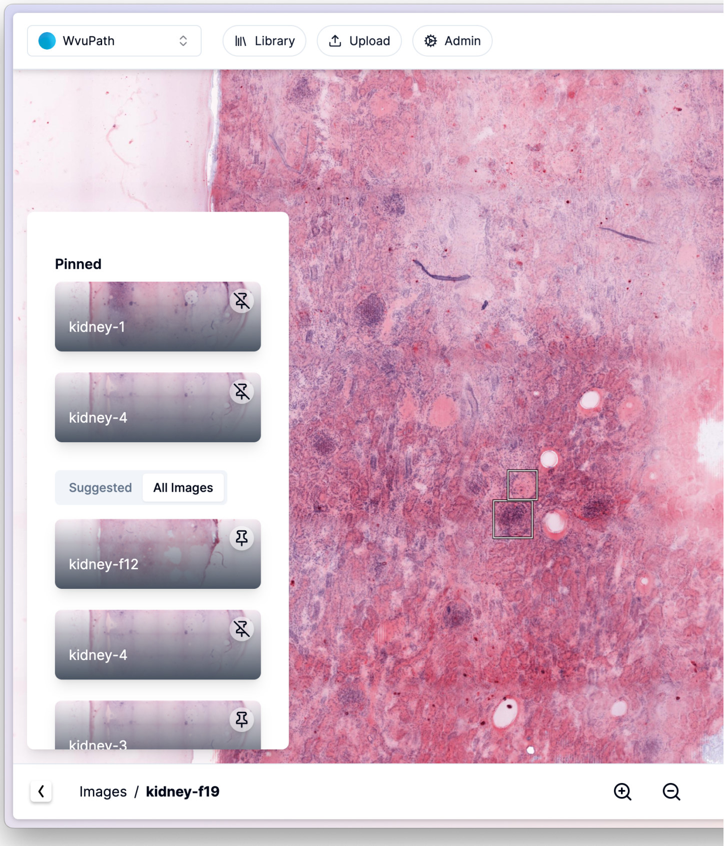
- Users were confused by whether image recommendations would change based on every image they visit along a chain of navigations (eg. A —> B —> C) vs. just be set once on image A. My resolution, from a UX perspective, is to fix the recommendations at A, since users don't want the entire suggested flow changing at each step along the way, making it hard to go back from B —> A in the future. However, that may not always be the correct desired behavior. Compared to the cognitive load of explaining and then offloading that decision to the user on a per-image basis though, I think the inconvenience of going back to All Images and then reselecting Image B in order to regenerate the suggested images is a (barely) acceptable trade-off.
- Some users were looking for a way to navigate to two possible places: the first being All Images, and the second being Search Results. This is a difficult request, because it reflects latent need and potential problems: assuming users even understand the difference between the two, and disregarding the fact that one can get to All Images by going to Search Results and then just Resetting the filters, it means that users are looking for a way to navigate to the general image library and on to their next image without going through a dedicated switcher. Some considerations:
    - Having both an All Images and Search Results is pretty unacceptable. It would be a fantastic point for confusion, since they're nearly identical, distinguished by a Reset button. Well, then, we'll add a Reset button!
    - Since we don't want to encapsulate that button within the switcher itself (user testing indicated that users simply didn't find that button at all!) a better place would be to put it where users would look for it. We thus added a Library page to the navigation bar that lets the user navigate back to the Library of the organization that is currently active, in the same way that they might open that organization's settings, or Upload page. This isn't a perfect solution though — it can still potentially feel hidden away (poor discoverability) and has poor indication of what it will do (since the other options in the navigation bar open popups, but this one allows you to go back to the Library page, they may be conceptually and structurally similar, but feel like entirely different experiences for the end user, unified bewilderingly behind a similar interface).

We'll go with this design for now! At this point, I had (quite literally) less than 24 hours left with Instapath, so after wrapping up my user interviews, I moved on to implementing the decisions made above.

High Fidelity Designs







Part 5: Conclusion

Since I was short on time, especially as the internship came to an end around the time I was wrapping up this feature, I made some design decisions that I think were suboptimal, but where acceptable choices given my constraints and need to balance multiple interests.

This design features:

- Adaptive design that is simple by default and powerful as needed. Most users turned out to use Search Results, so the image switcher, by default, displays just that. Further interaction reveals the auto-suggested images, or the ability to pin images, but you can get by just fine with the default simple behavior.
- Preserves FOV.
- A design that makes way for multiple observed kinds of user flows.

However, I'm still convinced that:

- There must be a more elegant way to implement a discoverable switcher panel that preserves FOV and context, with a visual signifier to indicate its existence, and a keyboard-driven interaction for power users.
- There must be an elegant and intuitive way to convey the information about the library, search query, and image name, without resorting to a conventional file hierarchy mindset (eg. Library > Search > Image), as well as navigating between the layers without having to "reset."

Case Study 3: Navigation

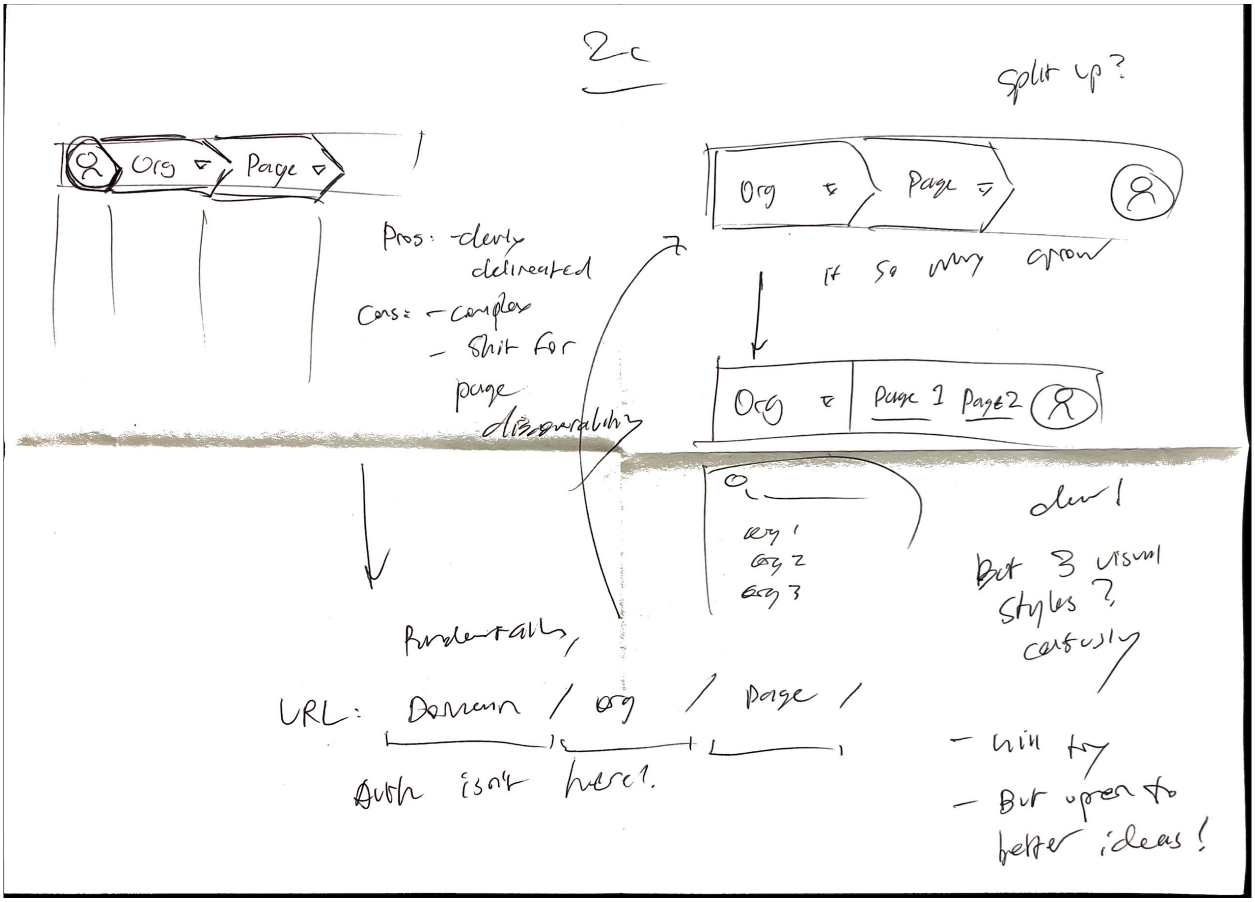
Let's now tackle a comparatively modest problem and keep it accordingly short. What should the navigation bar look like?

The problem is that navigation within the application is constrained by three factors: a) the currently logged in user, which determines b) the organizations that the user can view, and c) which page within any organization the user is on.

Hierarchically, we can express the path as User / Org / Page. How do we make this intuitive for our end users?

Part 1: Explorations

DSA doesn't have any concept of organizations. In their mind, every organization should host their own custom instance of DSA — a wildly difficult proposition to put into practice. With no precedent, we're kind of free to forge our own path here. A successful design here would be one that has a low time-to-page: dropping the user into any arbitrary organization and page, they should be able to get to a specific action fairly fast.



Some Lo-fi Sketches.

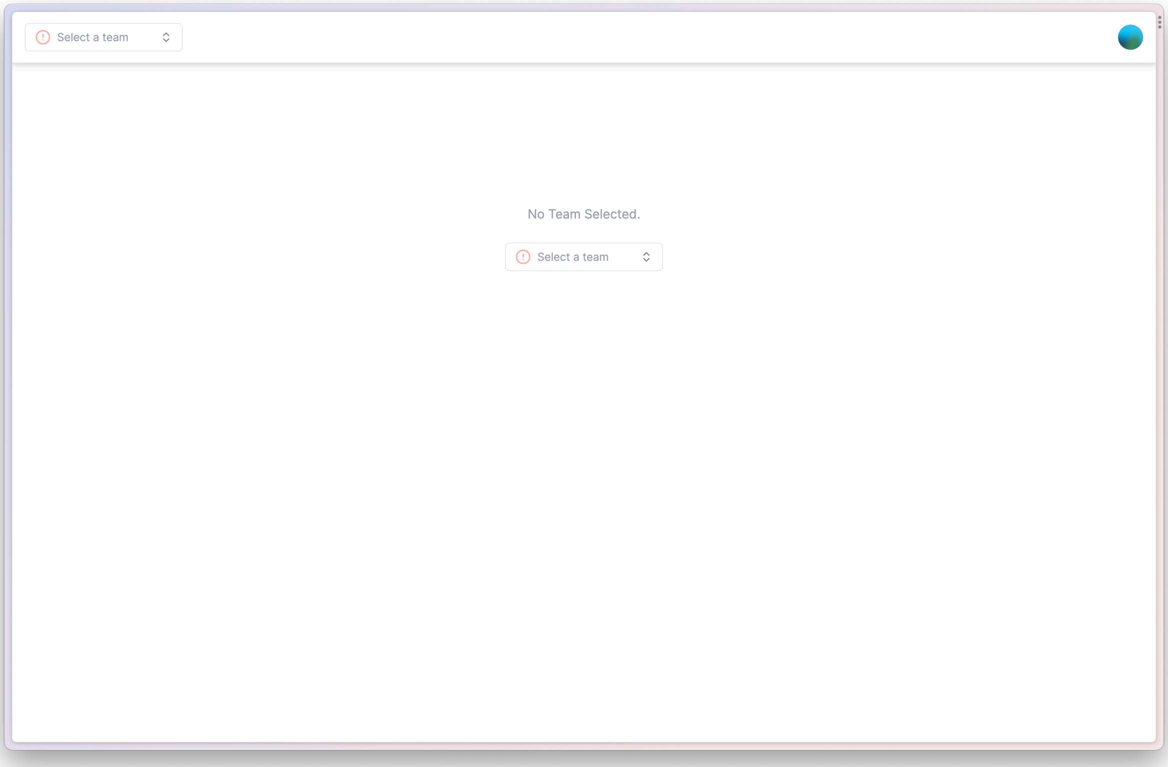
The first design I considered (top left) would be a three-layered hierarchical dropdown. From left to right, the user would be able to select the current user, the current org, and then the current page. Changing any option would reset all selections to the right of the change, and preserve all the selections to the left of the change.

As a design, it’s confusing and non-intuitive for users. The hierarchy forces the user to internalize and memorize the underpinnings of the UI, instead of reducing that cognitive load for them in the first place. The similarity between the UI for Org selection and Page selection could also prove difficult — especially when both are reduced to arbitrary user-determined names.

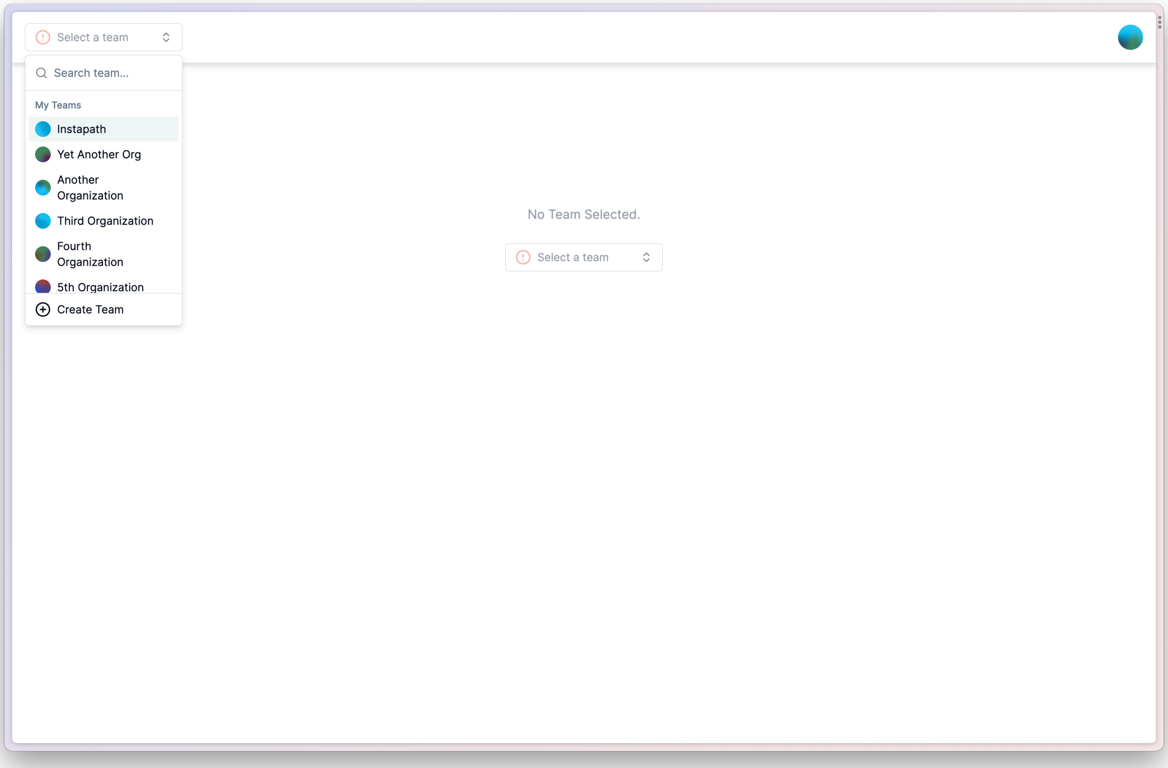
The second design I considered (top right) moves the user profile to the top right, separating it fundamentally from the hierarchy of the other two options. This is because, in order to switch between users, I realized that a fundamentally different flow has to occur: the user should log out, and then log back out. This also lets us put more robust user information and options (such as User Settings!) under the user’s avatar as options. However, I soon realized this doesn’t solve anything significant, not least of which is the fact that the similarity between Org and Page is only amplified now, and that the hierarchical design itself is the problem — not the number of levels in the hierarchy. I need to convey the distinction between options in different levels in the hierarchy without actually making the interface hierarchical.

Finally, I considered (bottom right), which puts the organization in its own dropdown, and the pages as fixed options on the right as Links or Buttons, with the User Profile in the same position as in the Second Design. The goal: to distinguish between the levels in the hierarchy via different visual styles, not explicit positioning. This is the design I then chose to go with.

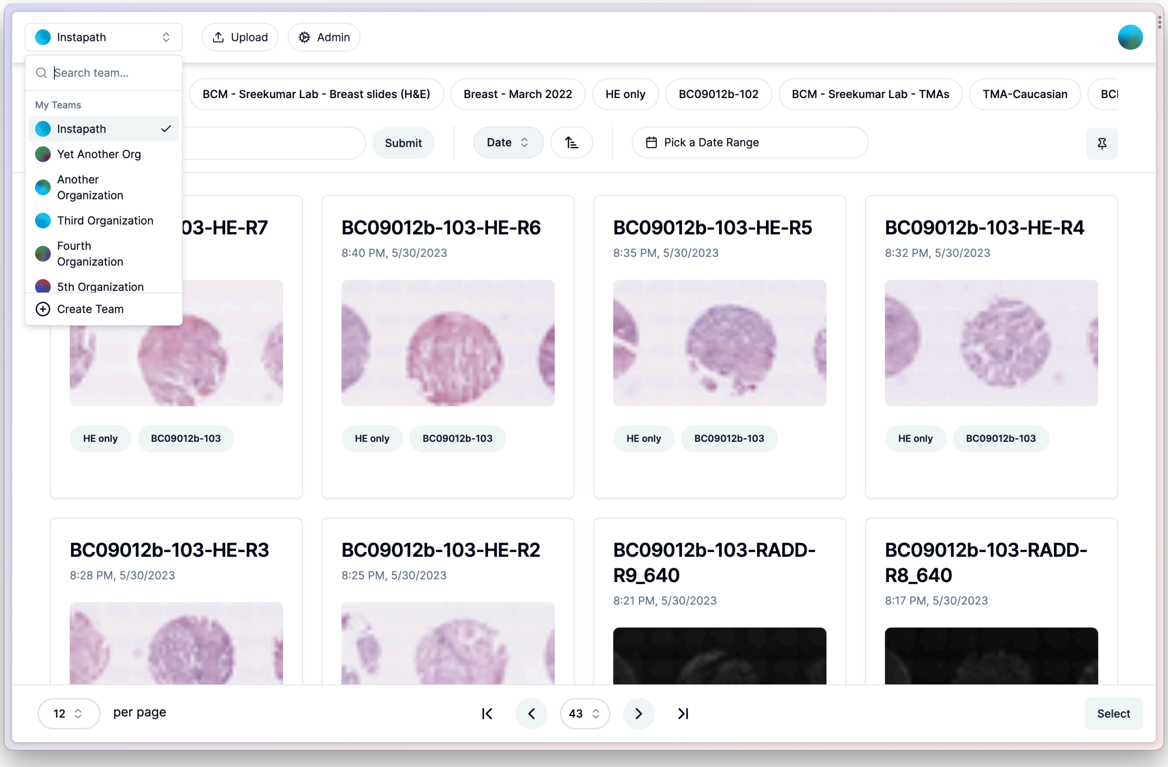
Part 2: Designs

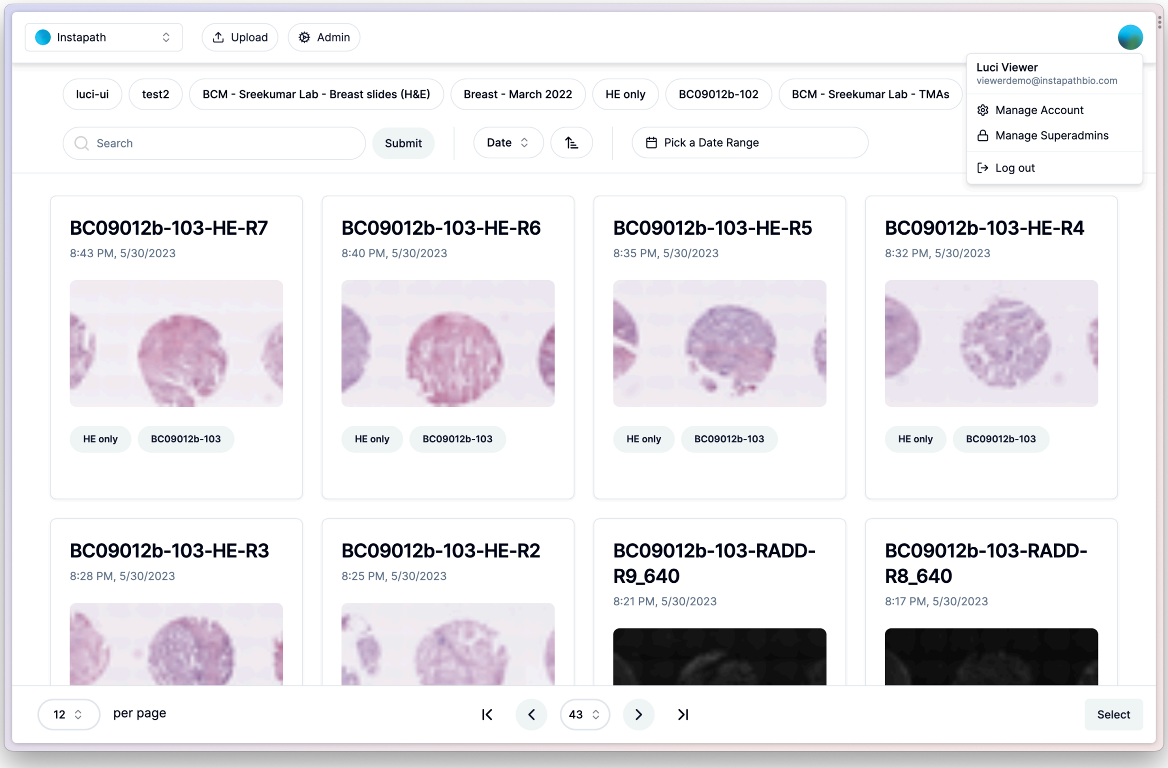


No Team Selected.



No Team Selected.





BC09012b-103-HE-R7
8:43 PM, 5/30/2023
HE only BC09012b-103

BC09012b-103-HE-R6
8:40 PM, 5/30/2023
HE only BC09012b-103

BC09012b-103-HE-R5
8:35 PM, 5/30/2023
HE only BC09012b-103

BC09012b-103-HE-R4
8:32 PM, 5/30/2023
HE only BC09012b-103

BC09012b-103-HE-R3
8:28 PM, 5/30/2023

BC09012b-103-HE-R2
8:25 PM, 5/30/2023

BC09012b-103-RADD-R9_640
8:21 PM, 5/30/2023

BC09012b-103-RADD-R8_640
8:17 PM, 5/30/2023