

UI Layout Planning

Saturday, September 18, 2021 11:14 AM

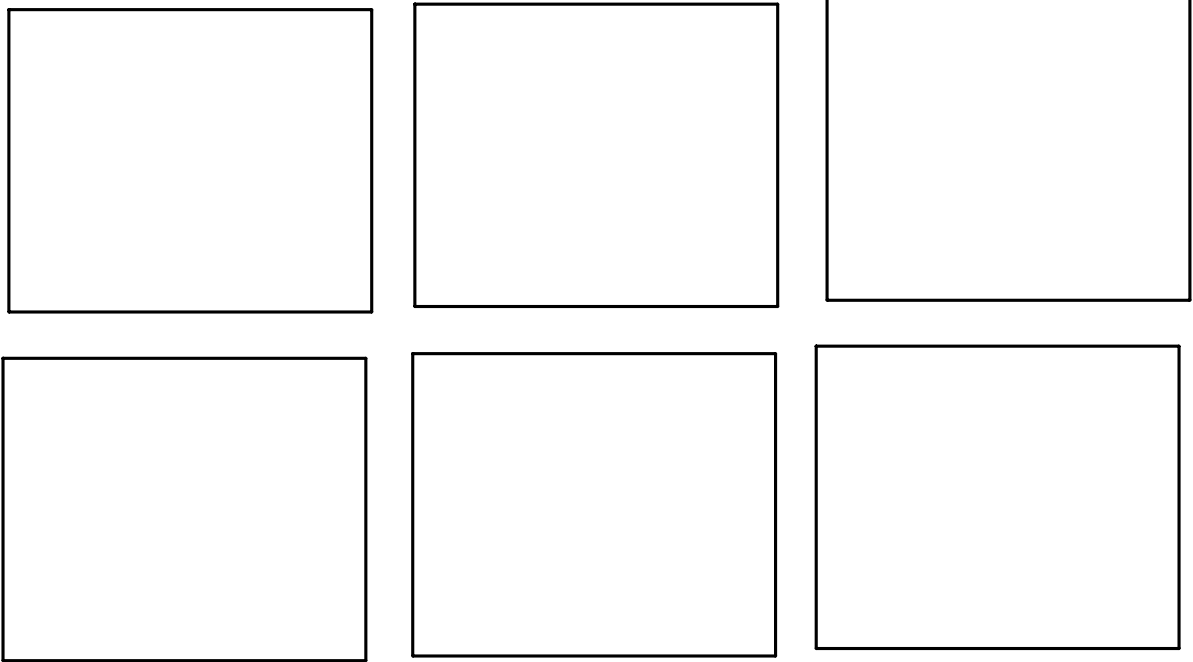
Image scrolling pane with each image having its own interactivity bar

Interactivity Bar: likes? Likes and dislikes? Happy, middle, sad?
5 emotion scale?

Main profile bar: pfp, account name, hb - bio, name, pronouns, etc. Also button for the user's own posts (opens second pane below)

Recommended users (also based off of similarity of content, not on the basis of who else likes the same people as you)

User's own posts pane



Each user involves:

- The image ids for their own images plus timestamps to account for recency
- Store the images they have seen as a hash map with t/f (basically just add to hash if seen, if not seen it won't be in map - so default of false)
- A unique user id
- User ids they are friends with

Access Points for data:

- Add new image:
 - o Image is sent to gcp function as a b64 the moment it is added onto the platform
 - o Function returns a url to the gcp bucket image link (decoded and stored as png)
 - o URL added into the collection of images with the UID of the user attributed to the image
- Nearest image url:
 - o Accepts image url of a given image and a user's UID
 - o Encodes the image data from the url and sends to henry's api
 - o Takes the returned url list
 - o Check through the given UID's to get what images a user has already looked through (basically see if each url in the set returned by Henry is contained within the hashmap)
 - o Return the first image which satisfies this constraint
- Delete image:
 - o Send image url that you want to delete
 - o Delete the image from the images set - no need to update anything else
- Get user images:
 - o Receive UID from request
 - o Query all image urls where UID is equal
- Mark as seen:
 - o When a user clicks right and opens up an image, that image should be marked as seen for that user
 - o Store the doc id of the image within the hash map as true