**Notes on the Project**

Stage 1 – Associations and Mappings

* Struggled with drawing out the associations
  + Tried typing things like “how do I draw out join tables for mapping of has and belongs to many in Rails?” etc
  + Looked at images and text.
  + <http://stackoverflow.com/questions/15597594/rails-mix-up-join-table-name-in-has-many-belongs-to-many>
  + <http://guides.rubyonrails.org/association_basics.html>
    - the immediate above is extremely useful.
* Methodology
  + Write it out on paper loosely.
  + Then consolidate in excel. (Active Record source very useful for this).
  + Then use Draw.io to make it neat and do it online.
    - Click advanced, then entity relationship and you can draw it out.
* Question:
  + Join tables
    - If it is a story has and belongs to many categories
    - And a category has and belongs to many stories
  + Should it be one table?
    - Or two separate ones
      * Ie category\_stories
      * And story\_categories
* Emily on Slack:
  + *once you’ve created those tables with those columns you need to:*
    - create join table for categories\_stories
    - create join table for locations\_stories
* The only time you need a join table is for a many to many relationship. So if a story can belong to many categories, and a category can have many stories associated with it.
* Question:
  + If you name something in the server side is that what the front end will see or does it not necessarily correspond to that?
  + I.e. in the database it will be user\_id for stories. But I would prefer the user\_name and tag\_line to come up on the client end.

**Websites/ apps to take inspiration from**

* Yik yak
* Instagram
  + (For Profile Part)
    - Could do it by time/ location / jointjokers (JJ’s)
  + Search
    - Jokers/ Locations/ Vibes/ Time
* Twitter
* Facebook

Go onto notes and just do devise (notes are from instagram example) for users (so that they can log in and log out). Then go back and see Alex to create the database.

Gem to search things

* <https://github.com/activerecord-hackery/ransack>

To DO next

* loop through categories and locations on stories.show (stories/id)
* create log out on menu (just like the instagram example)
* create seed data for emoji’s
  + add some default locations to begin with
    - could do this a different way later using “Google input” but for now
* and then look at adding comments to a story
  + see lesson plan in notes (from Slack)
  + In the directory “acts-as-commentable”
  + Also “acts-as-votable” (to give something likes)
    - Gus has slacked it to me.
    - <https://masteruby.github.io/weekly-rails/2014/08/05/how-to-add-voting-to-rails-app.html#.Vjh6Mq7hA0R>
      * do this one first as it is easier

Questions

* When doing data in the seed file there must be a quicker way to do the numbers in ascending order (eg cat1, cat2, cat3 etc.) like in excel.

Emojis

* Just realised that there are better images if you click on them which are bigger and a higher resolution. Should I change this for the future? Guess should just wait until see how it turns out.
* Got from this website
  + <http://emojipedia.org/>
* quite jokes to have fun different things with the emoji’s in terms of adjectives
  + eg peach could mean peachy etc.
  + so therefore semiotic as well as material.
  + Play on this binary.
  + Or w⚓ = wanker
    - So could do some fun puns.
  + Or helicopter icon could mean chopper
    - Could suddenly develop a whole new trending syntax/ associative meaning to the English language.
      * Anecists could bring chaos to the English language and disrupt it with its very own lingo/ slang/ language.
        + The language of jokes.
  + This is a social media company.
    - Why me?
      * Self professed social media addict.
      * Completely hooked
      * And I think this is a problem.
        + “Fuck Facebook”

join the anecist revolution.

<https://github.com/HenrikJoreteg/emoji-images>

* could potentially fork this and then git pull it and you get loads of emoji’s instantly.
* Storify

<https://storify.com/>

* Buzzfeed
* Medium.com