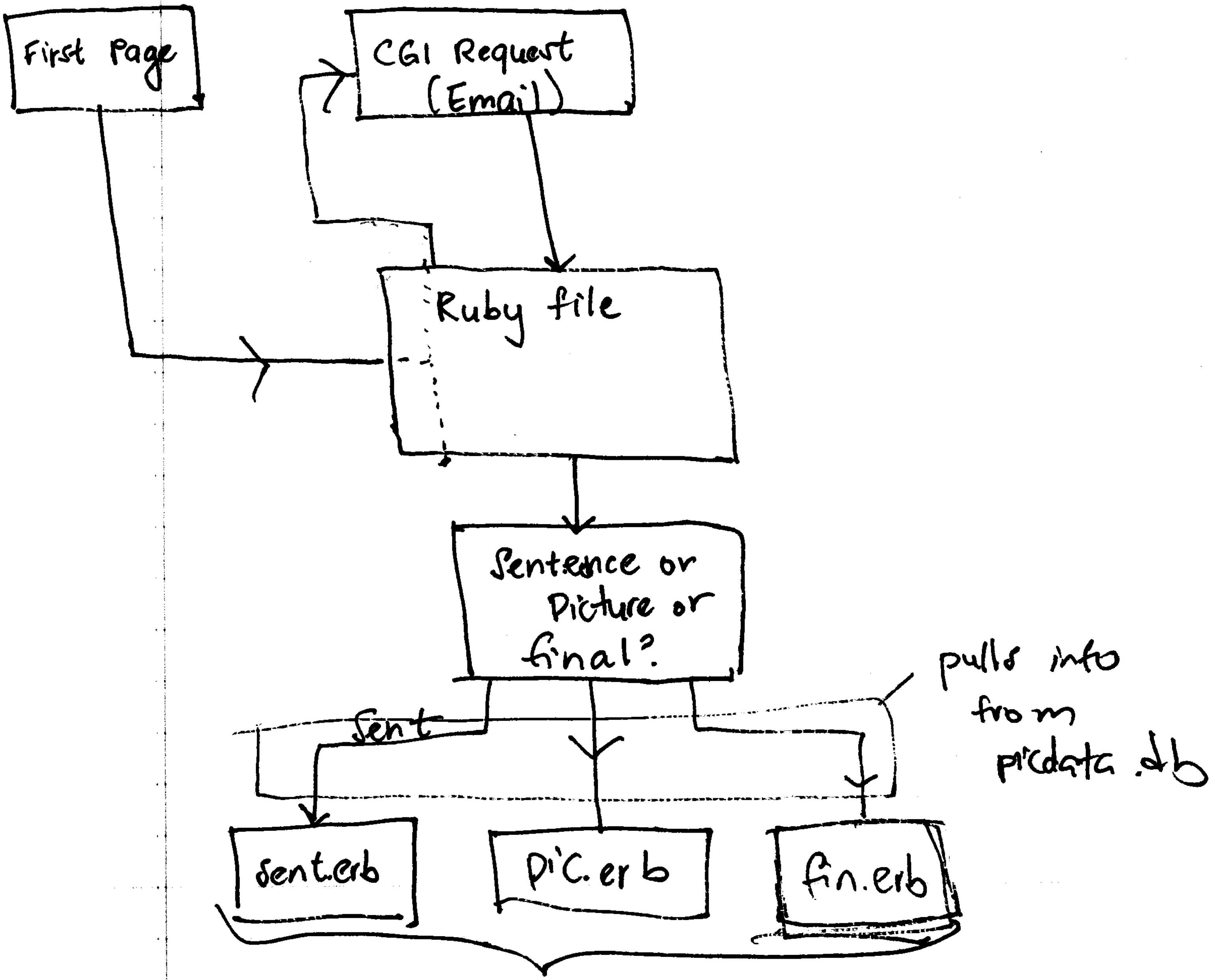
stuff to do:

- Make an email fynction
- Use token (make sure token matches)
- Have a way of opting out
- ~~filter out~~ bad email addys / ~~first in .js~~ then on web server
- ~~prevent SQL injections~~
- ~~write function that displays everything~~

- two boxes (one for first player, one for friends)
- NFS - has hidden directories that are one layer up; put pzdata.db in there

- templates
  - final showing: ERB can do repeats over an array
  - in general: can concatenate templates

token in email AND in show!!



Common to all: draw.js?
picdata.db

```
if create = 1
  - make new game ID          - ~~a~~ update game player ~~tab~~
  ✓ search for player ID, add new player ID > check valid e
  - add new drawing, make drawing ID
  - current turn # = ~~0~~ 1
  - send email to second player
        - it includes image (link, or just include) (opt.)

→ if (~~email~~ player clicks on link) & (turn # is even)
    - fetch image from previous turn & display
    - show empty div where can write sentence + submit
          ~~after~~
  <-if submit >
        - add new sentence row to database
        - email next player
        - update current turn id.

same
→ if (player clicks) & (turn # is odd)
    <submit>

c (player clicks) & (turn # = # players)
  after submit, show all previous sentnces + images
  mail whole game to all players

        - updating database
        - submitting
        - emailing next person
        - displaying      - final diplay
```

~~[illegible]~~

~~Class Game {~~

~~- Obje - list of Players~~

~~-~~

~~Class Player~~

~~- email~~

~~- turn #~~

~~- picture~~

- Email when turn ready

- Can do most fetching of game state in ruby

```
puts <<EOF
<dir ... >
   <      >
EOF
```

} ~~separate~~ file

Game #

/game.rb?gameid=x & turnid=y & token=z

Passin' as param

(create

Token

Games ID	Current Turn #	
1		
2		

Player ID	Email	Opt of Out
1		
2		
3		

Game ID	Player ID	Turn #
1	1	1
1	2	2
1	3	3
2	1	1
2	~	2

Drawing ID	Data	Game ID	Turn
1			
2			
3			

" " for sentence