

Alex Trahan, John Popich, Osayd Esmail

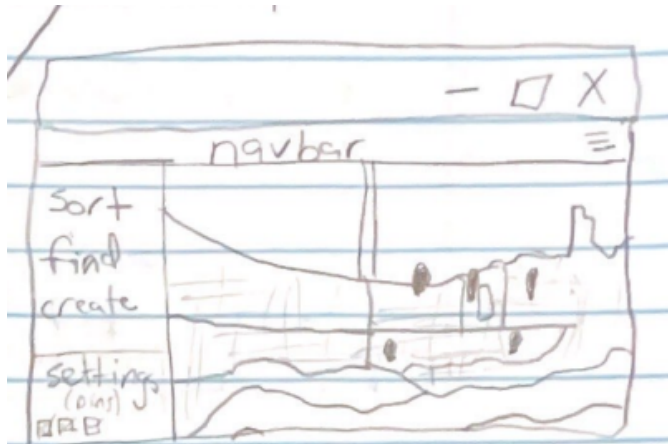
## Work Divide

Hopefully in this document we are able to 100% divide what is server's responsibility & client's. Once that is complete, take a chunk of work. We should go into pseudocode development.

- **Frontend:**

- HTML library: **NAME/Alex**

- mapView.html (div)



- Homepage: **NAME**

- GET from server ejs lib... html

- AccountMgmt: **NAME**

- GET from server ejs lib... html

- Login:

- Submit → email, pw → <https://ip-api.com/> →  
to server
- Create Account:
  - Submit → email, pw → <https://ip-api.com/> →  
to server
- Map: **Alex**
  - GET pin collection from server... server from db
  - Map will render client side
- MapMenu: **NAME/Alex**
  - Find:
    - Return random pin on map
    - FindClickedFromMap:
      - Display info for pin clicked on map
  - Sort:
    - Brings up checkbox visibility menu:
      - "Yours"
      - "Friends"
      - "Categories":
        - "Niche"
        - "Oddity"

- "Haven't Seen This"

- Create:

- Brings up pin creation menu:
  - "Name": pin name
  - "Location": pin's lat,long
  - "Categories": pin's categories

- Nice, Oddities, Haven't Seen This

- "Comments": user's comments

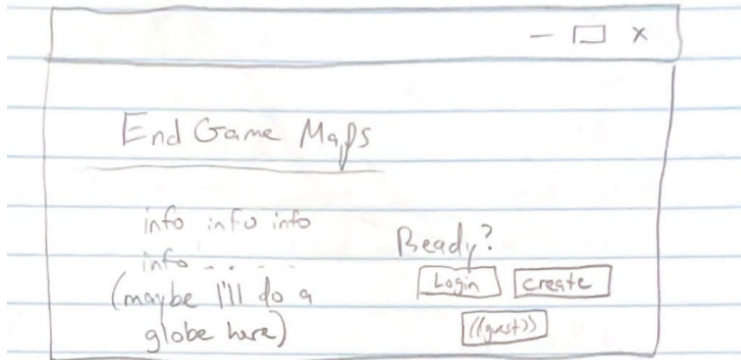
- Manage:

- Brings up list of user's pins
  - List item, list item, ...list item...
- When clicked on pin in list → bring up the editing & delete options for that pin.

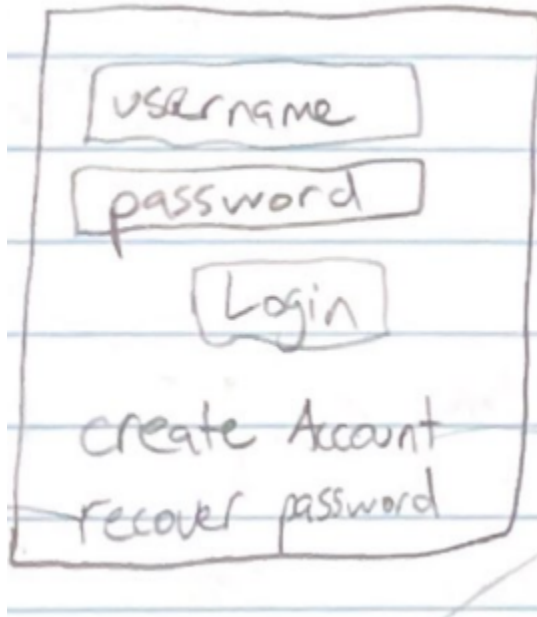
- Backend:

- EJS library: **Server rendered html data**

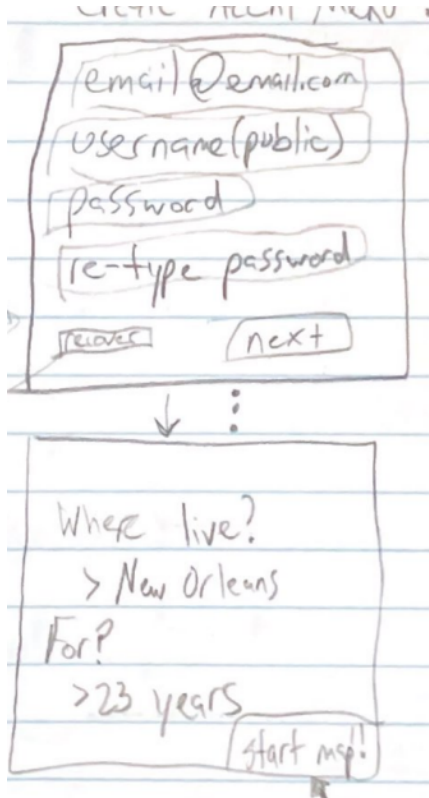
- Homepage.ejs:



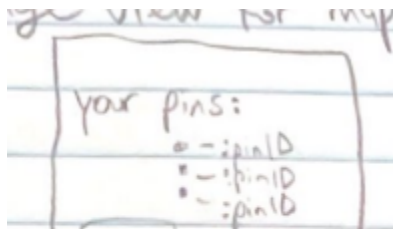
■ LoginMenu.ejs:



■ CreateAccountMenu.ejs:



■ pinManage.ejs:



■ pinManageUserPin.ejs:

name: name  
location: loc  
Categories:  
☒ ☒ ☐ ☐  
Comments:  
text here from before  
Save  
DEL

- Routes: **NAME/Alex**

1	/account/loginMenu/	Gets loginMenu → LoginMenu.ejs
2	/account/createAcctMenu/	Gets accountMenu CreateAccountMenu.ejs
	/account/createAcctMenu/createAccount	Gets data from form; ask Sec questions
3	/account/createAcctMenu/createAccount/create	Gets ?City&?Length
6	/login/loginSubmit/	User Auth
7	/login/createAccount/	Redirect to #2

12	/mapMenu/manage/	Gets pin management menu, show current users pins → pinManage.ejs
12b	/mapMenu/manage/:pinID	Gets pin management menu for <i>that</i> user's pin → pinManageUserPin.ejs
13	/mapMenu/manage/:pinID/delete	Delete pin from DB
14	/mapMenu/manage/:pinID/edit/submit	Get user entered data, update data in DB

- **Database:**

- If one of you guys can do this, that'd be great because I'm unsure how to set it up for the easiest implementation.