**Lead Em**

Before Release:

* Calendar styling
* Overall search page styles (include <section>?)
* Search criteria return proper leads
* Search Form component should be a form
* Make form actually use FormGroup
* Calendar doesn’t clear the date when clicking ‘x’
* Fix styling on label
* Change Application.html to index.html
* Make detail page able to save
* Integrate markup for Details page
* Integrate landing page
* Modal functionality appears broken - fix
* Remove /edit from path to edit a lead
* Allow creation of new users – new component
* Integrate Markup into Lead Form Component
* Integrate IsActivelySeeking checkbox and Boolean property
* Integrate Notes field and text property
* Allow user to create a brand new lead
* Rework header to have a logout button that isn’t there unless you’re logged in, and the name of the user
* Make Logout work in the header
* Try using Mongoose AutoIncrement for integer Ids instead of string
* Lead Detail page and component
* Integrate Mongo
* Write tests
* Implement Users
* Make app land on sign in page
* Implement sign-in and sign-up features
* Implement session checking and tokens
  + Finish implementing middleware that validates tokens
* Make Lead Detail a form
* Apply validation rules
* Factor all controls out of the form
* Debounce Search
* Make Search by UserId works (test by putting userIds in the leads at loading)
* Save UserId on Lead document
* Make a decision as to how Neighborhoods will be handled now that Search and detail use the same form
* Make sure versioning (\_\_v) works – if there’s time
* Implement Neighborhoods in Search and LeadModel
* Put Try-catch blocks in the relevant repo methods
* Modal for success in signing up
* Make search fire automatically for all fields in the lead form (so far all text fields work)
* Style Login button and Auth form
* Put route guards to avoid having unauthorized access to detail/search
* Something is wrong with the repo method for search. It returns as many copies of each lead as there are cities selected in the search criteria – probably happening in the SortByNeighborhood method. Fix line 70 to ensure the lead doesn’t get added more than once to the list if two cities match its criteria
* Add Email for User Auth and make sure it is unique
* Add wait spinners for CRUD calls
* Aot and rollup compilation (issues: aot broke because of ‘/Common/….” Having to be replaced with “Common/….” In every component file, and rollup was saying guid import didn’t work in BaseModel (replaced with a random js Math number generator))

Bugs:

* Phone Number Missing
* Modals not showing up on CRUD calls
* LeadModel.Phone.AreaCode error
* Aot or rollup breaks modals and some CSS
* CSS Display is wonky on smart phone:
  + Fields in Lead form are wider than the screen resolution, should be 100% of screen width
  + Green lines are wider than the display, should be 100% of screen width
  + Search Dictionary is shorter than phone width, should be 100%
  + The two buttons in the landing screen are too narrow, should be 100% of the phone screen width
  + Login screen is all messed up. Login form should be 100% of screen width

Deployment:

* Make Deployment build script:
  + Produces Dist and Aot folders in Client
  + Cleans all TypeScript files
  + Installs npm on both Client and Root
  + Creates production build folder with this structure:

leads.care

|

|-client

|-aot, dist folders

|-server

|-All js files in their regular folder structure

|-install

|-firstInstall.sh, Install.sh, uninstall.sh

|-node\_modules (root)

|

|-Main.js

* Make deploy script
  + Installs Node 6.10.3
  + Installs Mongo
  + Installs pm2

Milestones:

* Deploy to AWS or Digital Ocean:
  + SSH key for droplet
    - In command line:
      * ssh-keygen -t rsa
      * file to save in: /Users/Nicolas/.ssh/leadDroplet
      * cat /Users/Nicolas/.ssh/leadDroplet.pub to verify that the pub key is there, and to copy what this command returns to paste it into the SSH field requiring it. Copy the contents of that ssh file using the terminal command pbcopy < ~/.ssh/<ssh file name>.pub. (copy the whole thing, including the MacBook – Pro.local and ssh – rsa parts)
      * Paste this into the SSH key content field and in the name field, write in leadDropletSsh for the name
      * Ssh-add ~/.ssh/leadDroplet, give the mac password. – This is for Filezilla to log into the droplet. Because of this step, there is no need to add a keyfile in FileZilla
      * You can delete and create neww ssh keys on a droplet here: <https://cloud.digitalocean.com/settings/security?i=f09016>
  + Create the droplet (LeadDroplet)
  + Once it’s created, you have the ip address: 165.227.82.109
  + Password has been reset to qazplmQAZPLM135!#%
  + To SFTP into the instance, use FileZilla:
    - Host: 165.227.82.109
    - Port: 22
    - Use SFTP protocol
    - Logon Type: Normal
    - Password: Blank (because we ssh-add’ed the leadDroplet
  + To ssh into the droplet:
    - Simply go to the mac bash terminal and enter ssh [root@165.227.82.109](mailto:root@165.227.82.109)
  + Added new user: with ‘adduser nick’
    - Name: nick
    - Pw: iojknmIOJKNM90()
  + Gave the new user sudo priviledges:
    - Usermod –aG sudo nick (the ‘-aG’ stands for append group, because we are appending the sudo group to the nick user)
    - To test that this worked, you can enter id nick, and in the groups, you should see sudo (27) as one of the groups this user has
  + Steps to disable password login, and use ssh exclusively (for nick user only, not root)
    - Generate ssh keys on the server:
      * Mkdir ~/.ssh create the ssh folder
      * Chmod 700 ~/.ssh make it accessible only to our user
  + The app was moved to /home
  + Install pm2 globally: sudo npm install pm2 –g
  + To start the app using pm2: pm2 start Main.js “appname”

Future:

* Make checkbox label check/uncheck a box
* Make BaseComponent that includes modal functions, form functions
* Style modals with animation, refactor to use our own framework
* Make decent caching system
* Make LeadForm into a reliable pattern with associated API that is repeatable
* Datepicker and checkbox array should be its own component (neighborhoods)
* Use angular materials
* Fix modal problem with aot compilation
* Make Node/Yeoman-based app scaffolding CLI to automate app scaffolding and asset addition. In other words, you can add an asset the way you add a component in Angular-CLI, except adding an asset will make you the client-side a

TCR after first demo:

* Cache search parameters (works except Move-in date and neighborhoods)
* Clear search criteria button
* Make read-only text phone number field
* Make lead rows collapsable with content
* Add edit button in detail page
* Make content editable when edit mode is on

Issues often encountered after not touching the app for a while:

* If you’re getting an error when accessing the LeadsSearch page, where the content returned is { title: “error” }, the problem is in the server-side TokenValidate middleware. Make sure there are no errors in there like arguments not having a type and such.
* If GitHub desktop won’t include the Client folder, it is because Angular-CLI creates a .git and .gitignore resource on creation. Once you delete these two, you should see the insides of the Client folder show up in the GitHub desktop
* If you get the node side stopping at emitDestroyScript in async\_hooks.js, you’re likely trying to reach the server with a CRUD request with an expired token. Log out and restart the app

How to get the app running after a while of not running it:

* Run mongod
* In the top leads directory, build TypeScript (shift-cmd-B)
* In the Client folder, run ng build
* In the browser, go to <http://localhost:3000/#/>, this is the login window