**Ship**COMPLIANT  
**Licensing Experience Redesign Process**

**1: KICK THOSE TIRES!**My first step in recreating the Licensing experience within the Ship Compliant application, was to take a long stroll through the existing ShipCompliant SAAS app, to learn more about the current user experience and analyze the applications personality, if it had any to preserve within the Licensing experience. In doing so, I was able to bring continuity and at the same time, fully overhaul the entire Licensing experience. This was mainly done by shaking loose, the existing and painfully dry Licensing experience, and then creating a newer, lighter and easier Licensing experience.

**2: WHO USES THIS (cr)APP ANYWAY?**Due to my time constraints I did not have time to organize proper surveys or user interviews to create personas so, after creating a list of questions and emailing them screenshots of the existing Licensing experience, I called all of the friends I know that work within the wine industry,most of them located in California, and interviewed them each. Using the information gathered from my “user-interviews”, I was then able to generate a solid list of existing features to preserve but improve upon, missing features to be added, and existing features that were harming the user experience and needed to get tossed.

**3: WHAT TO KEEP & WHAT TO JUNK?**Next I leveraged any of the main site’s current Styles and/or UX patterns, that I thought still worked and were relevant, and revised those ever-so- slightly to accomodate a mobile/responsive grid pattern and semantic framework. This was for future use, when the SAAS platform will be fully shifted over to a mobile/responsive framework, which they’ve since implemented).

**4: ABSTRACT OR DIE!**  
I quickly learned my number one challenge was the massive amount of data, within the licensing application, and how visualize it best. The overall aesthetic demanded simplicity and intuitive organization, to prevent the user from arriving and feeling overwhelmed by the huge amount of data. The best way to do this was by showing the least amount of info, at any one time. This was achieved through the use of: a collapsing navigation, tabs, accordions and tool-tips, to aid in preventing the user from becoming so overwhelmed, they quit the application. I then addressed any glaring issues with existing functionality (ie. lack of intuitive data sorting/filtering options, the ability to click and select multiple states at one time, etc).

**5: THE STICKY-ICKY!**

Next, on sticky-notes, I wrote down and color-coded all of the features that would remain, that would be created and that would get junked: **Existing & Remaining *(Blue)*, Non-Existing & Implementing *(Orange)****,* and **Existing & Deleting *(Pink)***.

**6: SOOOOO SKETCHY!**  
From here I began sketching and iterating on the new architecture and overall layout of the licensing section. At this point in the process, I would normally seek feedback on my sketches and sticky-note features list, from my client & team, in order to iterate on them accordingly, based on their feedback. Being a one-person-team, I was dependent on catching any issues here myself, before moving on to digital wireframing, using Omnigraffle and ultimately a semi-functional prototype, using Azure. As a result, I spent a lot more time in the sketching phase than I normally would have; this helped me fix mistakes far easier than if I’d made it to the prototyping phase of the process.

**STEP 6: SPRINTING TOWARDS THE FINISH LINE**

After my sketches of the new layout and lo-fi wireframes were iterated on, and greenlit, I then created a hi-fi wireframe, using Illustrator. This was revised and greenlit, giving way to the creation of a semi-functional prototype, using Axure (CSS/HTML). After iterating on the new prototype, I was then able to write detailed, task-based stories for Ship-Compliant’s front & back-end developers, using Jirra to organize and manage their progress and git,to file check and keep things tidy for the developers.

**WHAT I BROUGHT TO THE TABLE:**

During my research and annalysis of Licensing data, I saw how helpful it would be if a user were able to filter the returning data based on variable such as: lowest/highest cost, ease of acquisition, availability, or the time it takes to get license issued.

**ASSUMPTIONS I’VE MADE:**

* I assumed that conversion rates will not be affected by omitting the service fee from the price listed
* I assumed a made up date for states not currently ranking in the most popular list. The data I display here, is as placeholder only.

WITH MORE TIME I WOULD HAVE:

* Pushed specs to functioning and mobile/responsive prototype with a bootstrap framework, to test the intuitiveness of the newly revised IA and UI (on multiple devices).
* Flushed out and revised the design much more, including custom icons, new checkout page, redesigned footer etc. The majority of my time was spent addressing the glaring information architecture issues and generating solutions. (Agile practices generally support the “form should follow function” rule, especially when there are time constraints).
* Spent more time researching other possibly useful features that I hadn’t thought of.
* Performed A/B and other relevant user-centered testing on the newly revised Licensing section of the app and iterated the prototype based on the resulting feedback.

**OTHER SEMI-RELEVANT RECOMMENDATIONS:**

Make a short fun animated demo that demonstrates the ease of use, usefulness and affordability of your 4 main tiers of services. Maybe capture their info at the end of the demo and offer a free month to test out the software? Right now it feels like a bait and switch when you click the demo button now and then are forced to enter your info. One thing you could do is measure analytics and bounce rates of people viewing that demo page and then see how many people actually fill out the demo form vs those that leave without giving their info.