AMNE

Feature Proposal:

‘Double Dip’ Realty

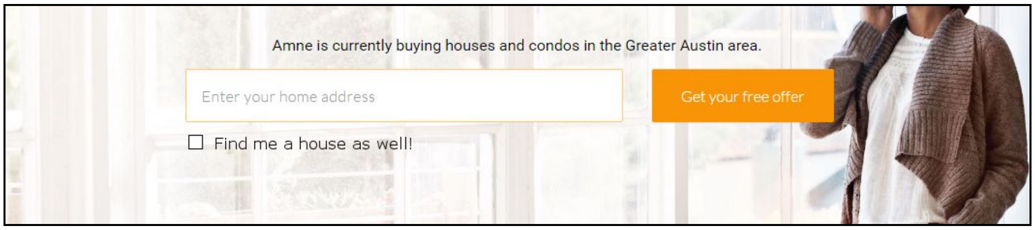
Vincent Wu

AMNE

>> Proposal >>

Oftentimes a client will want to sell their house and relocate to a house in the ***same*** area for financial reasons. Whether downgrading or upgrading, there is certainly potential for Amne to participate in both sides of the sales equation. I propose an additional feature that implements this in the initial form as an option (see image), and handles the necessary implementation in the next phase of the pipeline.

We want to prioritize sales of houses that Amne already owns for the best return of profit. However, if none are available, we want to go with the next best option, which is getting a cut from a realtor OR acting as the realtor ourselves.



Amne is currently buying houses and condos in the Greater Austin area.

☐ Find me a house as well!

>> Tools & API >>

1. **Oracle SQL database** - Database of Amne's internal property listings AND cached listings. For argument's sake, let's say this is an Oracle SQL database.
2. **SimplyRETS** - An API for Multiple Listing Services (MLS) is necessary. My research has led me to SimplyRETS, a RESTful API that normalizes listings from various MLS into JSON formats. The issue currently is that even if we manage to secure partnerships with various MLS providers, we will end up having to write hundreds of interfaces with all of their various formats.
3. **Google Maps API** - We want to use the maps here to mark points on the map that we have listings for. Ideally, internal listings are prioritized in order, perhaps.
4. **Standard web tools** - Bootstrap, JQuery, AngularJS, Flask, etc. Not critical for high-level.

>> Process Overview >>

1. User submits forms, prompting an HTTP POST request.
2. Redirect user to loading page
3. Using the address provided, server client receives request, sending a query to the database for internal listings and cached listings, and if no external listings are cached, sends an API request to SimplyRETS, caching the results for future use.
4. Normalize this data into JSON key-value blocks, and load new page, with listings as the variables.
5. Use Google Map's marker API (see here) to mark the listings on the map and load the UI.
6. User now has access to a UI with that has three regions: a map region with listing markers, a list of clickable listings, and a region for 'more info'. See padmapper.com for an example.
7. The user then uses the contact form (presumably through PHP or client-side code) to contact our partnered (or internal) realtor if they are interested.

Additional Notes

- Caching MLS API calls is crucial. These will no doubt be expensive in some way or another, so we want to minimize them as much as possible.
- Prioritize results from our internal database. Possibly only expound further if user clicks on 'more listings', or we do not have enough to fill X amount of listings.
- We can also make an additional crawler feature to scrape listings from Craigslist and other sites *where this is not illegal* (eg. not Zillow).