

THE PLACEBOOK

APARTMENT FINDER



Carl Collins, Amy Grude, Matt Scholl, Robert Thompson

Real World Problem	3
Justification for Need	3
Overview of Solution	4
Scenario	7
Design Choices & Rationales	8
Apartment Pins/Markers	8
Overlays	9
Criteria Selection	9
Criteria Ranking	10
Saving	11
Testing	12
Strengths	13
Shortcomings	14
Shortcomings of our system include the following:	14
Future Testing	14
Questions	15

REAL WORLD PROBLEM

For undergraduates, especially those coming fresh to a University, the choice of housing is a simple one - the Dorms. Even as they are forced out of the comforts (and madness) of controlled housing there is ample lodging for them in co-ops and divided houses close to campus (often called the “academic ghetto” due to lack of upkeep on the part of the landlord and neglect on the part of its young tenants). If, on the other hand, you are moving to a reasonably sized college town for graduate study, you may be more discerning in your choice of lodging. Searching for housing is a complex problem that balances perceived needs and personal preferences along with some very real constraints (including but not limited to budgetary constraints). Being new to town can add further wrinkles to the process. In such cases, decisions on housing often precede opportunities to familiarize oneself with the area or, for that matter, to establish contacts with those who could offer advice. If you don’t yet know your way around, how do you know if an apartment is in “a good neighborhood” or “close to a grocery store”? Shouldn’t there be some kind of system to better support this task?

JUSTIFICATION FOR NEED

Using Ann Arbor as example, we can imagine someone doing a simple Google search for “Ann Arbor Apartments”. The top result, thanks to the ability to pay for sponsored links, is McKinley.com, a national real estate firm whose corporate office is located in Ann Arbor. McKinley’s site relies on external links to map services to give a sense of location, even though it is not that hard to make internal use of the Google or Yahoo Maps API if one spends a bit of time and resources. Also McKinley is not a source for generalized listings but merely for those properties managed by the company. Users searching for apartments in Ann Arbor might also come across apartments.com, which relies on a more diverse pool of sources than the McKinley site. However, apartments.com is covered with advertisements for other services, and provides fairly standard and limited ways to select criteria (it gives users a series of binary check boxes). Once again any sense of actual location relies on an exterior map service displayed in another window. The user is forced to jump back and forth between the ‘search criteria’, ‘results’, and ‘map’ pages. With this kind of system users must rely on their own resources (such as a text editor or note paper) to make comparisons, thus making the sense-making process unnecessarily complex.

There is an abundance of choices for starting an apartment search, but each of these sources can be defined more by what they lack than what makes them stand out. In fact the abundance itself can lead to a sense of overwhelming choice - while still leaving a user with the impression that there is no good fit to their particular needs or preferences. Housing is particular in that there is often a fixed constraint such as cost or size as well as nuanced preferences that are difficult to capture with simple binary choices. Further the current services rely on text listings

which have a higher cognitive overhead for browsing than a more visual presentation might allow.

Based on the available tools, we determined that user centered visualization can and should be applied to apartment searching. The goal of our design can be divided into the following elements: integrate the visual importance of location, include relevant location data (local knowledge), make criteria selection more than binary (yes, no, maybe), provide immediate visual feedback as users modify their selection criteria, include demographic information like crime statistics, allow browsing based on specific locations, and reduce switching.

OVERVIEW OF SOLUTION

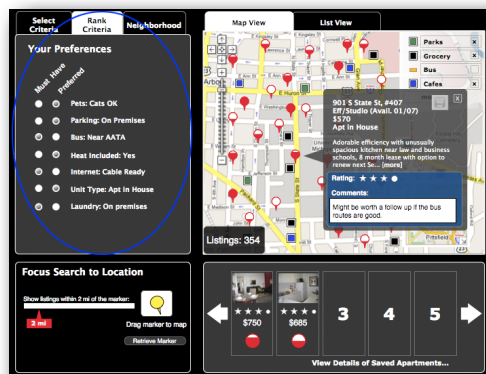


Our final design incorporates all of the disparate elements of existing apartment search tools as well as additional tools to reduce cognitive demands on users. Stepping through our criteria we see the following:

- ♦ The upper right hand part of the interface is a map, with additional layers of information beyond apartment listings



- ◆ There is a tab for neighborhood location and the ability to use the neighborhood map as a navigation aide



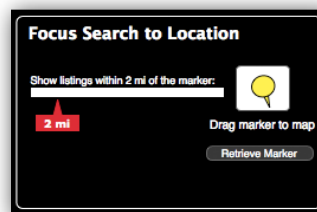
- ◆ Criteria, once chosen, can be ranked as a must have or preference
- ◆ Differences in how listings match criteria are easily identified by number of listings and through the icon of each apartment pin on the map.

- ◆ Changes to 'Must Have' criteria are immediately reflected on the map, as the number of apartment pins appearing on the map increase or decrease accordingly
- ◆ Changes to 'Preferred' criteria are reflected in the appearance of the pins themselves. In the image below, the center pin is completely filled, reflecting an apartment for which all preferred criteria are met; the pin on the right is not filled at all, reflecting an apartment for which none of the preferred criteria are matched; and the pin on the lower left is partially filled, which reflects that a small portion of preferred criteria are matched
- ◆ Crime stats can be included as a map overlay. Other overlays include local businesses of in-



terest and features like parks, all of which can be easily added and removed from the map

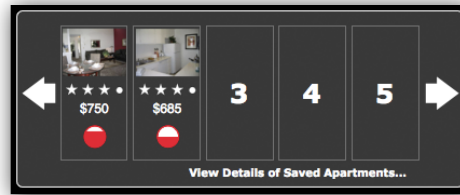
- ◆ The Yellow Marker feature allows a user to center on a non-definable location (like a building or road feature) and restrict choices to a given distance to this location.



All of the elements are contained in a single interface, different features are merely hidden behind tabs rather than separate components hopefully reducing switching.

- ◆ The number of units matching the current selection criteria is displayed in the bottom left-hand corner of the map, thereby providing users with another layer of feedback on the balance between their choices and currently available apartments
- ◆ Users are able to save 'favorites' as they discover them, thus making it easy to later revisit and compare favored apartments. This eliminates the need to maintain a log or journal in a separate application or on paper, thereby reducing the need to switch between applications or divert

attention.



A more complete walkthrough of our design choices can be found in section five.

SCENARIO

The user, a student recently accepted to the University of Michigan at a graduate level, goes to our service linked prominently through the UM housing site (or through an affiliate link paid for in Craigslist).

Consistently in interviews - price - is the first concern and most pressing constraint for people in the rental housing market. Accordingly, the first item to appear on the Select Criteria tab is Rent, for which the user is able to designate 'min' and 'max' values. Rent, Availability Date, Bedrooms, and Pets, all of which appear under the Basic Criteria heading, are treated as Must Have criteria. In other words, these criteria are treated as filters. Only those apartments which match all four Basic Criteria will be displayed on the map (where apartments are represented by red apartment pins). The Add Additional Criteria section allows users to select other criteria by which to constrain their apartment hunt. Criteria in section are treated as Preferred unless the user indicates otherwise. It is via the Rank Criteria tab that the user is able to modify the default Must Have and Preferred settings. The criteria listed on the Rank Criteria tab are generated automatically based on users' choices on the Select Criteria tab. The two tabs are dynamically linked, such that choices deleted from the Select Criteria tab are automatically dropped from the Rank Criteria tab.

If, for example, the user indicates (via the Rank Criteria tab) that Parking is a Must Have, the Parking parameter will shift be treated as a filter (rather than merely factor into the Preferred rating), such that only those apartments that have 'On Premises' parking will be shown on the map. Likewise, the 'Listings' counter on the lower-left corner of the map will display an updated number of available apartments. In this way, the user can easily modify the Must Have/Preferred settings for each parameter, and with each modification receive an immediate indication of available housing based on those selections.

When an apartment pin shows up on the map, the pin will reflect, by way of "filling up", how many of the 'Preferred' criteria are matched. An entirely filled (i.e. solid red) pin meets all of the Preferred criteria; a pin whose head is not filled at all (i.e. solid white) does not match any of

the Preferred criteria (though the fact that it appears on the map at all means that it has met all of the current Must Have criteria). Apartments which fail to match one or more of the 'Must Have' criteria are not displayed on the map. As the user updates the various Must Have/Preferred settings, the map updates dynamically, with pins displayed/hidden and their Preferred rating updating accordingly.

Ultimately the end results displayed on the map and list are a combination of filtering choices. For example: show me all houses with a price between \$800-1000 that have on-site laundry and allow pets, and maybe are within 0.5 miles of central campus (as indicated by the Yellow Marker feature), and maybe are in Kerrytown (a neighborhood, determined by local knowledge). Therefore, results in Kerrytown close to campus would show up with a fully filled marker on the map, whereas other matches that have the must haves but not the preferred selection would show up only partially filled.

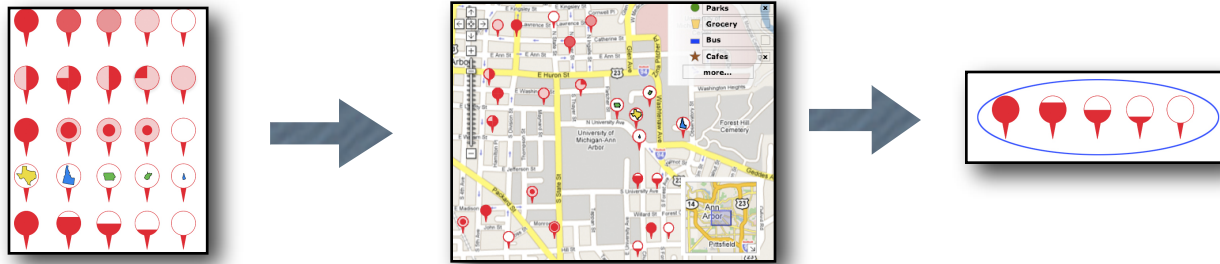
The ultimate data displayed on the map is a compound based on the filtering primitives selected by users.

DESIGN CHOICES & RATIONALES

Design choices were made through iterations of user interviews, design brainstorming, prototype development, and then gathering user feedback. Design choices, including initial and mid-process prototypes are shown below.

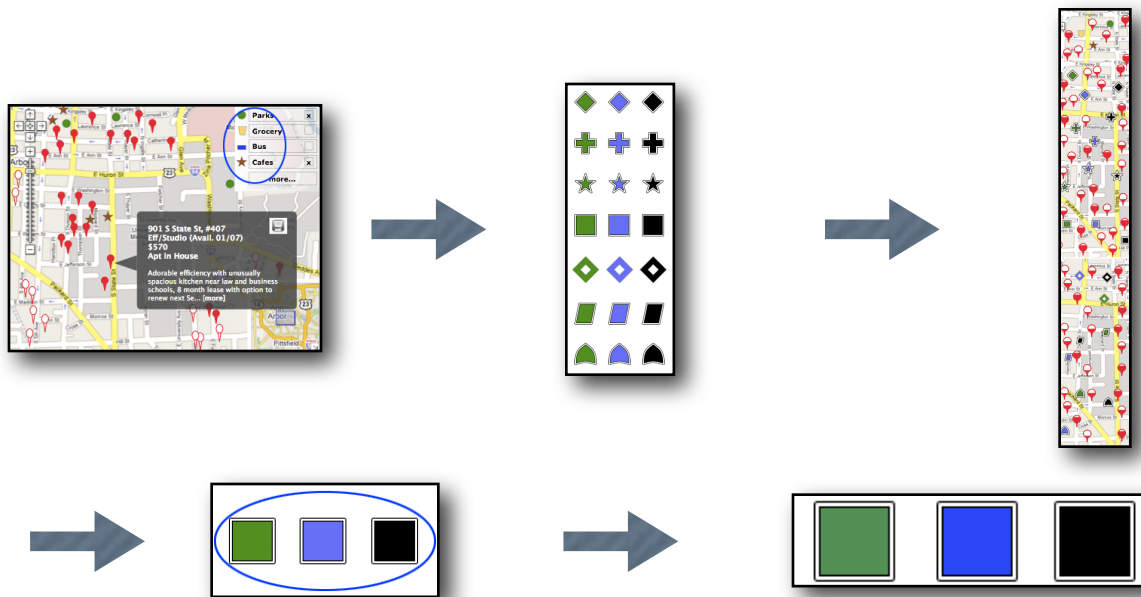
Apartment Pins/Markers

We generated five distinct designs for denoting an apartment location. Each design reflected a different approach to encoding the degree to which a given apartment matches a user's Preferred criteria. When initially looking at all five designs in a list (shown below, far left), each user liked a different option. However, when shown the markers on the map (center), users overwhelmingly chose the pins that "filled up" (far right) correlating to the degree of matching user preferences.



Overlays

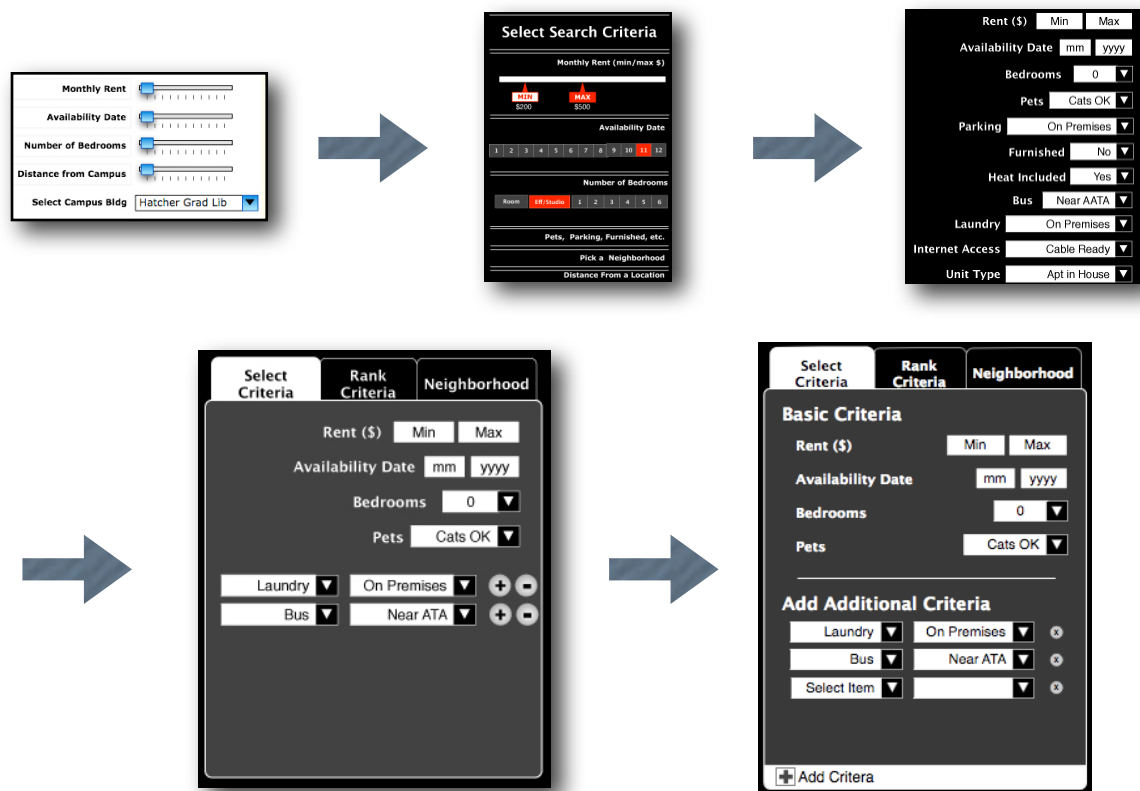
Similar to the pins, overlays denoting locations such as grocery stores, parks, and bus stops were tested using seven possible options, both in a list view and on a map. Once again, users formed a consensus when viewing the options on the map rather than in the list format. The square design was deemed the easiest to distinguish—its ninety-degree angles stood out against the round head and pointed tip of the apartment pins. Feedback about the colors of the overlays led to small changes in the final revision.



Criteria Selection

Criteria selection began initially with sliders. As the design process wore on, however, it was clear that the sliders took up more space than we had to work with. The next tactic was to try choosing criteria from a drop down list, but this strategy resulted in a seemingly overwhelming array of choices, as was true of other apartment finder sites we had evaluated beforehand (especially the Advanced Search option on the U-M off-campus housing site). The final design mimics the OS X Mail functionality of adding and deleting rules using embedded “plus” and

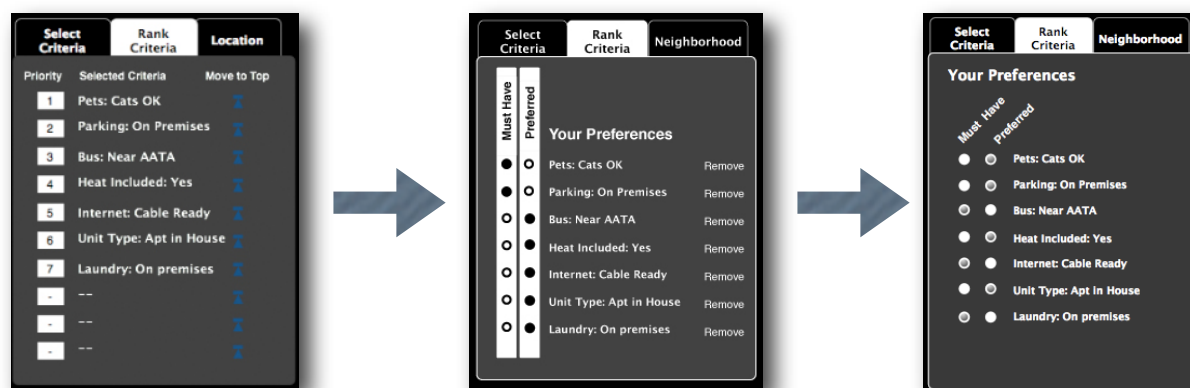
“minus” buttons. This format was altered to be more friendly for PC users, with only 1 “plus” button positioned at the bottom of the panel (with accompanying label, Add Criteria).



Criteria Ranking

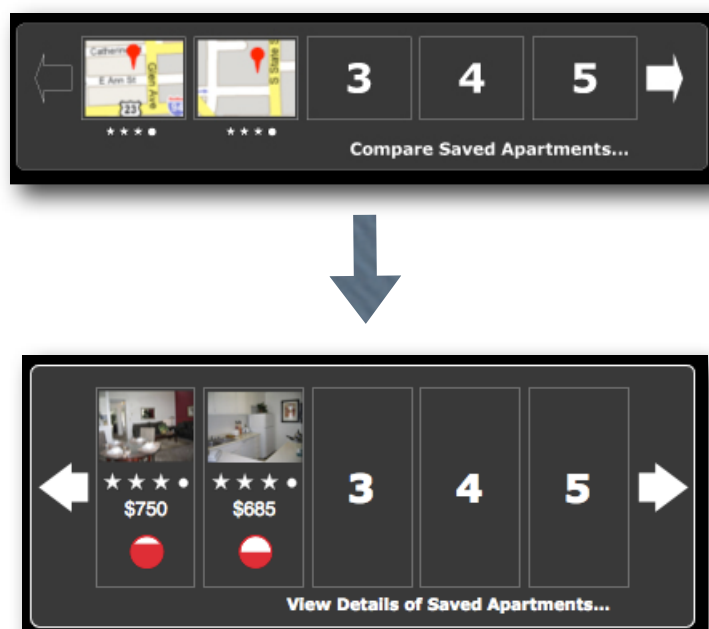
The origin of the ranking feature lies in the difficulty users have selecting criteria that give them a satisfactory result set. In all apartment finders evaluated, criteria were either “on” or “off.” However, from talking to users we discovered that although some criteria are “must have,” most were negotiable depending on availability and price. The criteria ranking panel allows the user to set their choices as “must have” or “preferred.” The result set will include apartments that meet all the “must have” choices, but may or may not meet the “preferred” choices. Initially, the ranking system was included in the same tab with the criteria choices; however, this feature was moved to a different tab due to clutter.

The first prototype (shown below, far left) required too much work from the user. Having only two options, Must Have and Preferred, proved to be an easier and more meaningful choice for users versus forcing them to rank their choices from ‘1’ to n. The strength of the white boxes (in terms of color and vertical orientation) on the second prototype confused users, who did not understand that the radio buttons were attached to a criterion next to them. The final mockup (shown below, far right) attempted to ameliorate that problem.



Saving

The ability to save apartments (for later review/comparison) was a popular feature with users; however, there were questions about how best to represent saved apartments. In the first iteration, saved apartments were displayed a panel underneath the map, which would display a pin in the context of the map location. Users did not find this helpful in remembering a unit; instead, they preferred an image of a room inside the apartment. Users also suggested adding the price to the saved apartments panel, as that was the most important aspect of each place they chose to save. Additionally, they wanted a way to rank the selection (on a 4-star scale). Along with the user's star rating, we have also included the top portion of the apartment pin, as indication of preferences met.

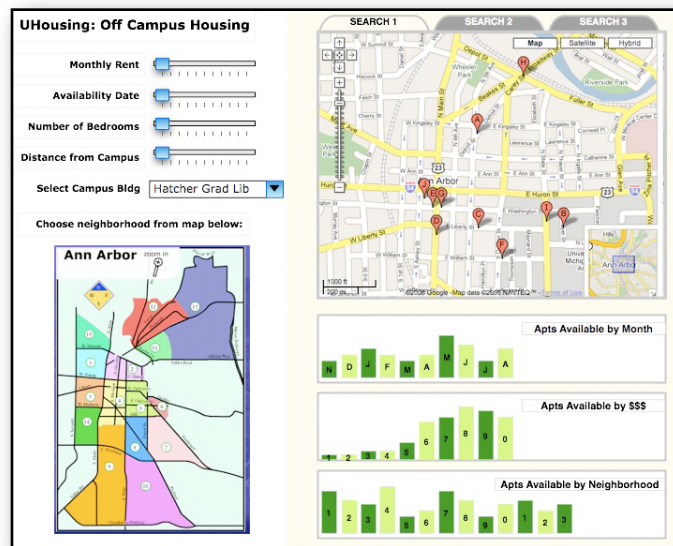


TESTING

For this project we relied on both interviews (to gather stories/scenarios of apartment searches) and tests of both printed and digital (but static) mockups. From the very beginning of our switch to the apartment finder we relied on user input to shape our design choices. Initial interviews gave us an idea of the existing field, as nearly everyone interviewed had conducted a search for an apartment online. An interesting trend in interviews was that while most people searched online, their final rental choice often came about as a result of a recommendation through a trusted friend or peer rather than an online listing. This particular aspect was not focused on because we felt that if a tool could be more reliable and able to provide more nuanced selection this might be overcome. Instead we focused on those areas where subjects felt that a given apartment search tool did what they wanted and those areas where it could have served them better.



Working from several concepts, based on early user interviews and examples from class lectures, we designed an initial iteration. This initial design relied on elements like sliders and histograms and tested poorly with users.



Going back over interview data and the responses to the initial design we began the iterative process that produced our final design. Throughout development we gathered reactions to our higher level choices. As a result of this, sliders of all kinds began to fade and more traditional web style drop down, radio buttons, and check boxes took their place - all as a result of user feedback.

As we began to finalize the high-level concepts of our system we turned our testing towards the individual visual components, chiefly the pins that mark rental listings on the map, the icons for overlays (parks, groceries, etc) and the overall layout of our elements. All of our final design choices are influenced by this user feedback and the results of these tests were critical to our decision making process.

STRENGTHS

The combination of features and visual innovations make this design stand out from currently offered products. When deciding what you want in an apartment our system lets a user only fill out as many criteria as are relevant. The ranking of what criteria you "must have" vs. "preferences" are critical elements to a better system. Current designs either have very few options (relying on search to constrain results) or force you to go through a whole page of options, which still might not contain what is important to you. Further these options are either always on, or always off. By allowing users to differentiate between their requirements and preferences the chances of finding a good match are increased. In terms of graphing out responses from a system the chance of false negatives is reduced. Further, by using visual indicators and visually displaying the level to which a given listing matches preferences we are not only creating a desirable user experience but allowing a richer context in searching.

SHORTCOMINGS

Shortcomings of our system include the following:

- ♦ Based on feedback from users, we are aware that Craigslist is a popular resource, not least because it is fast and easily browseable. While our design presents many useful features which are not available on Craigslist, our system would have to be fast in order for current Craigslist users to consider making the switch. Users would need to be able to quickly adjust apartment parameters and see those changes rapidly reflected in the map display. Short of this, adoption may be a hurdle.
- ♦ The social web is getting big - but is it useful? A significant percentage (but still less than half) of people interviewed seemed to indicate a final deference to references from people - if our system works well enough will it have greater trustworthiness than the existing systems? Are people going to always prefer something social and, if so, would some kind of social computing apartment finder be better? As is, we have considered adding a social computing component to the 'Neighborhood' tab, such that users could leave comments. This feature is currently out of scope, and as such is not discussed elsewhere in this report.
- ♦ There is a reliance on good information for our system to work. The mash-up nature of the backend data would require better and more consistent reporting of housing features and cooperation from a variety of stakeholders. Every rental agency has their own listings as well as listings in different sources, each with different information sets while there is little in the way of standards or consistency for listings from non-corporate listings. However if the system could be proved to be popular this is something that could be managed over time.

FUTURE TESTING

As a next step it would be wonderful to create an interactive version of this prototype, perhaps in Flash, and record reactions to features that are more animated than the static visuals we have already had users evaluate. Especially, we would like to implement the marker constraint. The idea of being able to label a point of origin (like a given building or physical feature, like a freeway onramp) and then constrain search within a given radius of that point is something that stands out as novel. We would like to conduct additional tests for this feature, given that paper mockups failed to provide users with a sense of how this feature might be used, or how it might shape their apartment finding task.

Generally speaking, the goal for future testing would be to move beyond reactions to components, as has been the case with much of our testing to date, and start gathering data on how tasks in the system are played out by users. Reactions to a more interactive version would be a great way to evaluate the overall visual nature of the system in comparison to the traditional

list based view (listings, though still, accessible are hidden behind the map) and could inform future development of similar systems.

QUESTIONS

- ♦ How much of this can be done given constraints of available information sets?
- ♦ How much of this is unfeasible from a development standpoint?
- ♦ What is the point of contention between visualizations, usability, established design models and user expectations?
- ♦ Does this aid sensemaking? Are there other components that might better aid sensemaking?
- ♦ How do comparisons change as criteria change? For example, if a user saves a listing, and then changes their criteria, should the saved marker reflect the new or old criteria?