Breadcrumbs Development Document

The Grass Is Always Feiner

Jonathan Balsano (jrb2194)
Logan Donovan (lrd2127)
Yanyi Luo (yl2504)
Mark Micchelli (mm3710)

Overall Process

Each team member came to this project with varying degrees of design and development experience. Jon and Yanyi have had significant experience with web design and JavaScript, and so they handled the bulk of the application implementation. Logan and Mark are both skilled programmers, but with relatively less experience with JavaScript and HTML, and so they created most of the documentation, while still contributing a little to the implementation. Our team worked together really well, because we clearly communicated our schedules at the beginning of the project, were motivated to learn the large suite of JavaScript tools we used, and all wanted to work hard to create a great product for the project.

The following was our meeting schedule, which we laid out at the beginning of the project, and strictly adhered to throughout the design process. In addition to the meeting schedule, our group was constantly in touch during all hours of the day (and night!) via email and text messaging.

November 26	Introductions; initial design discussions; review of Yelp bookmarks
November 27	More design discussions; presentation of initial sketches; first drafts of personas and use scenarios
November 29	Completion and submission of personas and use scenarios
November 30	Creation of initial and final markups on myBalsamiq; solidification of design ideas
December 2	Delegation of tasks; tutorials on project tools (Git, Backbone, Underscore, Mustache); start coding the implementation
December 5	Check-in meeting; further delegation of tasks
December 10	Long finish-up meeting; finish implementation and documentation together
December 11	Final meeting; video recording; presentation prep

Target Users

Breadcrumbs is designed for young adults, aged 19-26, who are familiar with technology and live in metropolitan areas. This target user class is largely composed of college students, recent graduates, or individuals of that age. Members of the target user class are typically social, and go out often, engaging with local businesses of a variety of types. Most frequently, they focus on restaurants and bars. However, this demographic also has a tendency to frequent music venues, theaters, and retail stores. These users are educated beyond a high school level, some beyond college. They have a working knowledge of technology and the Internet and frequently turn to such to solve day-to-day problems.

Persona No. 1: Jeremy the Connoisseur

Jeremy is a college senior attending school in New York City. He has a deep love for indie style, both in music and fashion. Jeremy is a philosophy major with an interest in film. In particular, he is a film noir enthusiast. In his free time he enjoys going out to bars with a few close friends, reading long dramatic texts, and playing guitar. His family is relatively wealthy, so when he decides to pursue something new, his parents usually foot the bill.

This year, being his final year in college in New York, Jeremy has made a mission of venturing to as many bars as possible before graduating, with the intention of trying a variety of different beers to refine his palate. He has thoroughly enjoyed exploring the bar scene in New York, as well as the variety of brews available, some local, others foreign.

Jeremy's Use Case

Jeremy would like to use Breadcrumbs to keep track of the bars of interest to him as well as the beers they serve. He opens the Breadcrumbs application, which he has been using for some time, and looks at the lists of places he has currently bookmarked. He currently has a few lists: "Bars I've been to", "Bars to try", "Lunch Spots", "Great Concert Venues", and "Restaurants with Beer". Today he'd like to finally sort out all the information he's compiled on each bar and restaurant to sort out where he can get the best beers. He starts by clicking on the "Add List" picture on the "My Crumbs" page, to create a new list called "Favorite Beer Spots". Then, Jeremy returns to the list menu by selecting "My Crumbs" in the breadcrumbs* at the top of the page. Next, he opens the list "Bars I've been to" by clicking on its image stack. He then starts scrolling down the list, viewing his notes for each bar by

---

* Please note that, as used here, "breadcrumbs" refers to the list of pages a user has been through to get to his current location, as often displayed at the top of webpages. This is a technical term that has no relation to our Breadcrumbs app. From this point on, assume that lowercase "breadcrumbs" refers to this list of pages, while uppercase "Breadcrumbs" refers to our app.

hovering his mouse over the "notes" icon underneath each business. If his notes indicate that he enjoyed the bar, Jeremy clicks on that bar's "lists" icon to see the lists that bar belongs to. To add that bar to his "Favorite Beer Spots" list, he simply clicks the checkbox next to "Favorite Beer Spots". When he's finished looking through his "Bars I've been to" list, he returns to the "My Crumbs" page by using the breadcrumbs, and then moves on to his other lists to repeat the process. Once he's done, he closes the app.

A few days later, Jeremy receives a call from a friend who says she'd love to visit on her way home from England when her flight lands in New York. Jeremy decides he'll have to take her to a bar to try some of the best beers he's found. Being that she's spent the past year in England, he thinks a bar with a great local brew would be the best choice. He turns to the Breadcrumbs app, goes to "My Crumbs", and opens up his new list of "Favorite Beer Spots". He then glances down the list of his notes by hovering his mouse over the notes icon underneath each bar. He spots a restaurant on which he noted "Best Brooklyn Beer I've had yet" and decides that's where he will take her. Having found the information he sought, Jeremy closes Breadcrumbs.

Persona No. 2: Amanda the Blogger

Amanda is a recent college graduate living in New York City. As a journalism major, she writes random bits for a variety of publications to make ends meet while she searches for a full time job. Amanda is a brunch-lover, the kind of person whose first response when looking to meet with someone is "let's do brunch!" She enjoys well-crafted food, and will often go out of her way to seek it out. Aside from her writing, she practices yoga three times a week, volunteers at a nearby animal shelter, and occasionally hosts dinner parties.

Currently, Amanda is working on a new project: a food blog. With all the brunch and dinner parties she hosts, and her expert writing skills, she feels that starting a food blog is a great way to showcase her talent and generate some buzz around her name. Unfortunately, she eats out so often that she can't write fast enough about all the food she's tried. Instead, she tries to pick the best and worst from each week and only write about those.

Amanda's Use Case

Amanda decides she needs a better place to keep track of all the food she's tried. Previously she's kept a journal after each restaurant trip noting what food she's eaten and a few short notes. Breadcrumbs seems like a great tool to use for this purpose so she decides to give it a try. Opening up the app for the first time, Amanda clicks on "+ Add Crumbs" to search for the restaurants she wants to bookmark. She sees the "Search Yelp" bar, and types in the name of her favorite restaurant in Greenwich Village. This brings her to a search results page, which shows her the restaurant she's looking for. Amanda notices the "add bookmark" icon underneath the picture of the restaurant, and then clicks on it to add it to the Breadcrumbs app. Amanda returns to the "Add Crumbs" page by clicking on the link

at the top of the app, and then repeats this process with her other favorite restaurants.

After Amanda pulls all of her bookmarks from Yelp, she realizes that her bookmarks are unorganized, with restaurants she's reviewed mixed in with ones she hasn't. Amanda decides she wants to organize all of her bookmarks into two lists, one for reviewed restaurants, and one for unreviewed restaurants. She goes to the "My Crumbs" page by clicking on the "My Crumbs" link at the top of the app, and sees an image labeled "Add List". She selects the image twice to create two lists, which she calls "Reviewed Restaurants" and "Unreviewed Restaurants." To populate these lists, Amanda selects "All Crumbs" from the "My Crumbs" page to see all the restaurants she's bookmarked. For each restaurant, Amanda clicks on the list icon, which opens up a section where she can add or remove the restaurant from her three lists: "All Crumbs", "Reviewed Restaurants", and "Unreviewed Restaurants". Once she's done this for all of her bookmarks, Amanda double-checks that her "Reviewed Restaurants" and "Unreviewed Restaurants" lists display the appropriate information. Satisfied, she closes the Breadcrumbs app.

The next day, Amanda decides to start knocking out all the restaurants she never reviewed, so she opens up the app to the "My Crumbs" page, and is confronted with her two lists. She selects her "Unreviewed Restaurants" list and browses through the list to decide which two she will review. Finding the brunch spot from two days ago, and dinner from a week ago, she keeps the restaurant information open on Breadcrumbs, and proceeds to write her blog post. When finished, she closes the Breadcrumbs app.

Design Decisions

The following are some important design decisions that faced our team. They range from core functionality concerns to minor aesthetic choices, but they all required thorough discussion and debate.

Levels of Categorization: Bookmarks, Lists, Tags, Categories, Crumbs?

During our first meeting, our team discovered a number of ways to organize business in Yelp. Yelp has separate sections for bookmarks, lists, tags, and search categories. As indicated in the assignment description, the organizational distinctions are not clearly laid out. Bookmarks have two built in filters, "to try" and "to review", based on what Yelp believes to be the most common use of bookmarking system. The user can also define custom lists into which they can add businesses, but it is not visually apparent or emphasized within the interface, more like an afterthought. The user can write notes about a business for their own use but it has no tagging system built in to combine this with larger organizational constructs. This means that in most cases the user ends up with a single collection containing all of the "saved" businesses in a single list, which can be filtered by the Yelp designated categories or by the two label options given to you.

At the beginning of our project, we spent some time trying to include all of Yelp's distinctions in our own app. In the end, however, we eventually decided in favor of simplicity. We wanted to make user organization very clear, easy, and visual, which is what we identified as the biggest weakness in the existing bookmarks. The Breadcrumbs app can pull businesses from the Yelp website to bookmark them, and then the user can sort these bookmarks into custom lists. All bookmarks are part of an "all bookmarks" list, and each bookmark can belong to multiple custom lists. This design was inspired by labels in Gmail. In Gmail, all emails are part of an overall "inbox" label, and then within the inbox, user can categorizes their emails using custom labels. We picked this example since it was something that we were all familiar with and though worked particularly well. As in Breadcrumbs, emails in Gmail can also have multiple labels.

But how could a user easily find a bookmark within Breadcrumbs? Here again, we found inspiration in Gmail. Gmail allows you to search for an email *within each custom label*, instead of only allowing you to search the overall inbox label. Similarly, our group decided to provide functionality for searching within only one particular list. In addition, we wanted to take advantage of the premade Yelp categories within the API. As a result, we're allowing users to filter their searches using these categories.

Our core design decisions were guided primarily by Nielsen's eighth heuristic: aesthetic and minimalist design. By paring down the number of categories we could use to categorize our bookmarks, we managed to create a fully functional, better-than-Yelp bookmarking system with only five distinct markups on myBalsamiq.

Designing the Crumb Icons

Our team wanted an easy way for the user to know the *complete* set of actions she can execute when dealing with a bookmark. This goal was inspired by Nielsen's sixth heuristic: recognition rather than recall. Rather than needing to remember which menu to go to for each task, we lay everything out in front of the user. Additionally, all of the icons have the same image and meaning no matter which page of the application you're on, be it the list view, the search results view, or the single business view. This goal was inspired by Nielsen's fourth heuristic: consistency and standards.

Edit Mode

Our team quickly realized that we wanted to allow the user to change the order of the crumbs within each list. We decided that, by selecting a toggle, users would shift modes from the default "view mode" to a new "edit mode", which would allow users to drag the business around the list to reorder them. Our "edit mode" was inspired by the edit mode on an iPhone or an iPad, where users also drag icons around to change the layout of their application menus.

A little later in the project, our team realized that we should also give users the ability to change the order of the lists within our main "My Crumbs" view. In this case, we almost decided to give each list a submenu that would allow you to move it around the list view. However, we eventually realized that this functionality is almost identical to the functionality within a list view. Following Nielsen's fourth heuristic—consistency and standards—we decided instead to create an edit toggle, just like we had before.

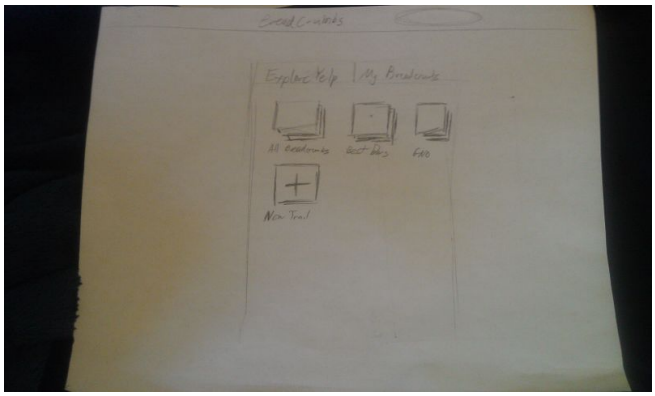
The Breadcat

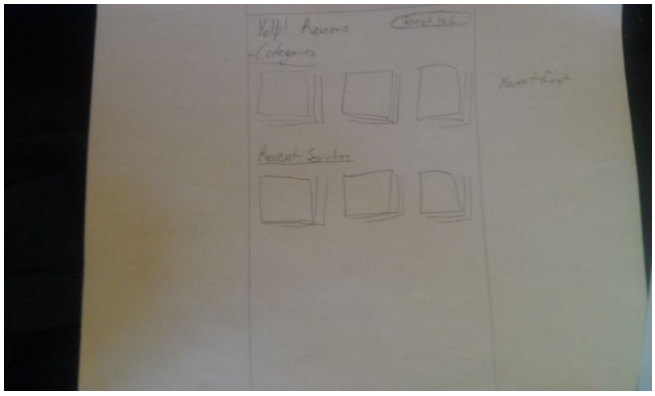
Throughout our application, we've included a little graphic of a cat stuck inside a slice of bread. This was designed first as a cute little mascot for Breadcrumbs. Later, we realized that sometimes our app took a little time to save and render. We decided that we should include a spinning breadcat in the top-right corner to indicate to the user that our app is loading. This was inspired by Nielsen's first heuristic: visibility of system status. Rather than leave the user wondering if she did the right thing, the spinning cat lets her know that Breadcrumbs is working on processing her request.

Prototyping

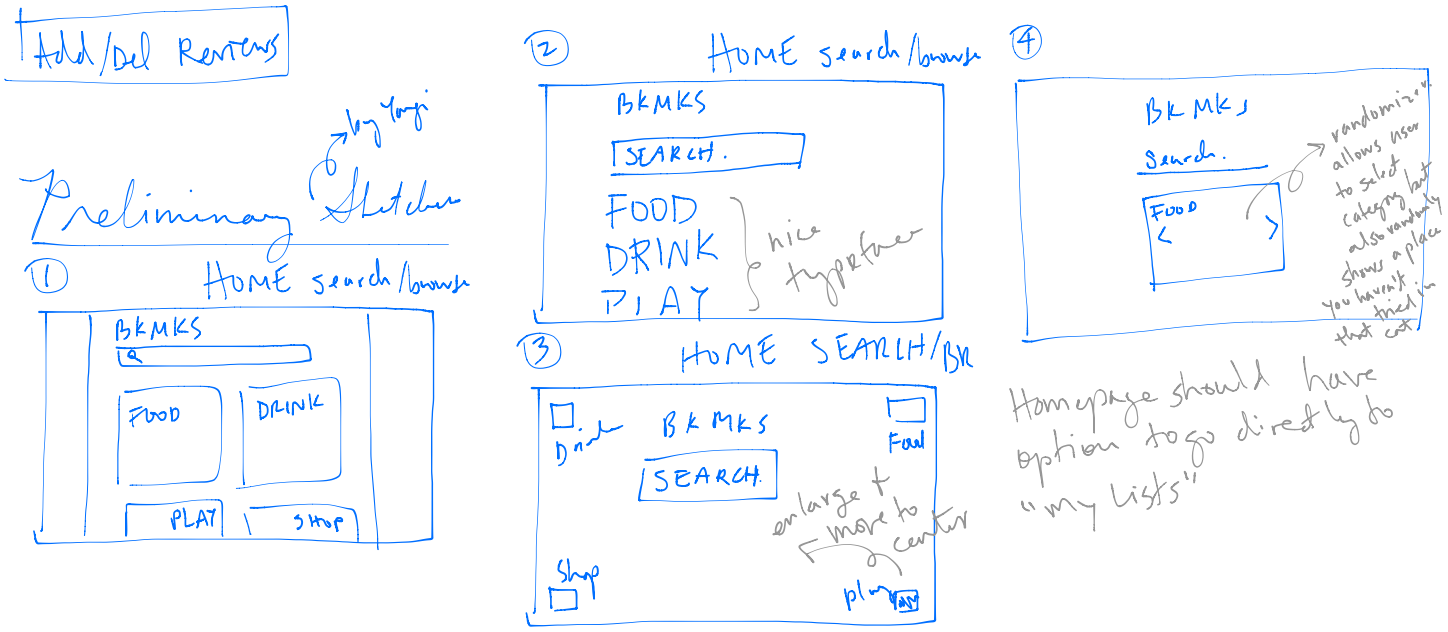
Our prototyping began with sketches drawn up by Jonathan and Yanyi. Remarkably, their sketches, created separately, turned out to be nearly identical. We then set to work copying the hand-drawn sketches into myBalsamiq. This initial foray into myBalsamiq familiarized each team member with the overall app design, and also highlighted early, obvious design flaws and mistakes, which were soon eliminated. After this step, our group took a short break, and then critiqued each of the five markups in turn. This helped us solidify our ideas, and after some debate and discussion with our group members, we ended up with a final design that we were all proud of.

Here are Jon's original sketches, which were ultimately made into myBalsamiq markups:

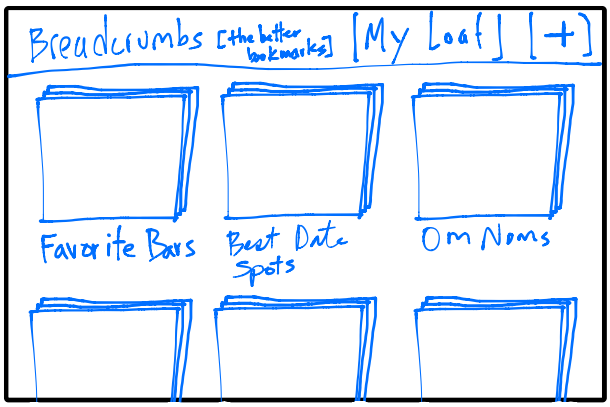


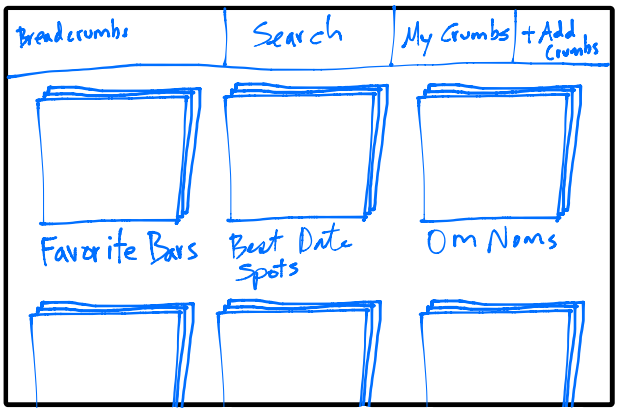


Additionally, here some of are Yanyi's initial sketches (made on an iPad). Below that are the initial and final versions of our myBalsamiq markups, with detailed comments.



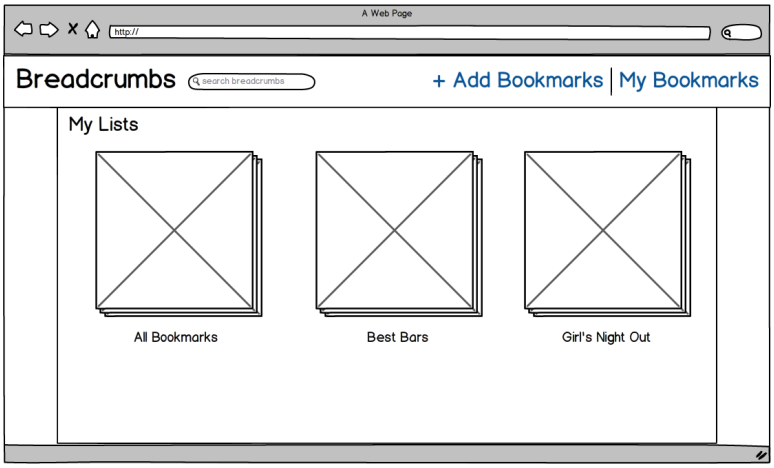
Search/browse existing bookmarks
Add bookmark from Yellp
Lists View
List View
Single Item View



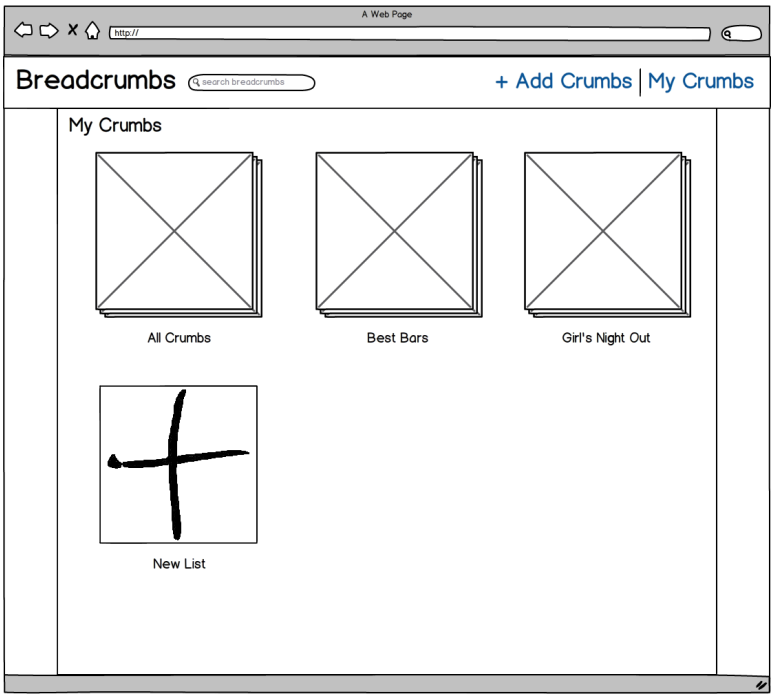


Lists View, or My Crumbs

This is the initial markup of the lists view, which appears whenever the user clicks on “My Bookmarks”. We decided here that the first list that should appear is the list of “All Bookmarks”, which is the master list that contains all business loaded into the app.

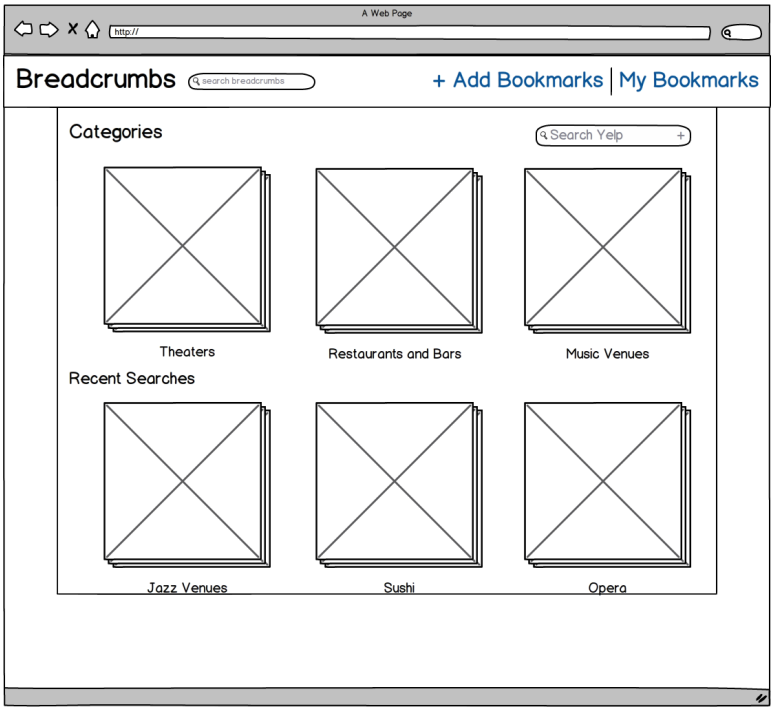


This is the final markup of the lists view. We wanted an easy way for the user to create a new lists, so at the end of the list of lists, we added a big “plus” icon to show the user how to add a new list. We also decided here to change the terminology from “Bookmark” to “Crumb”, in order to give the app more character and name recognition. This change is reflected in all subsequent final markups. Additionally, the “edit mode” toggle is not shown in this markup, because that was a later design decision. However, there is an “edit mode” toggle in the single list view, which ultimately we used as a model for this view.

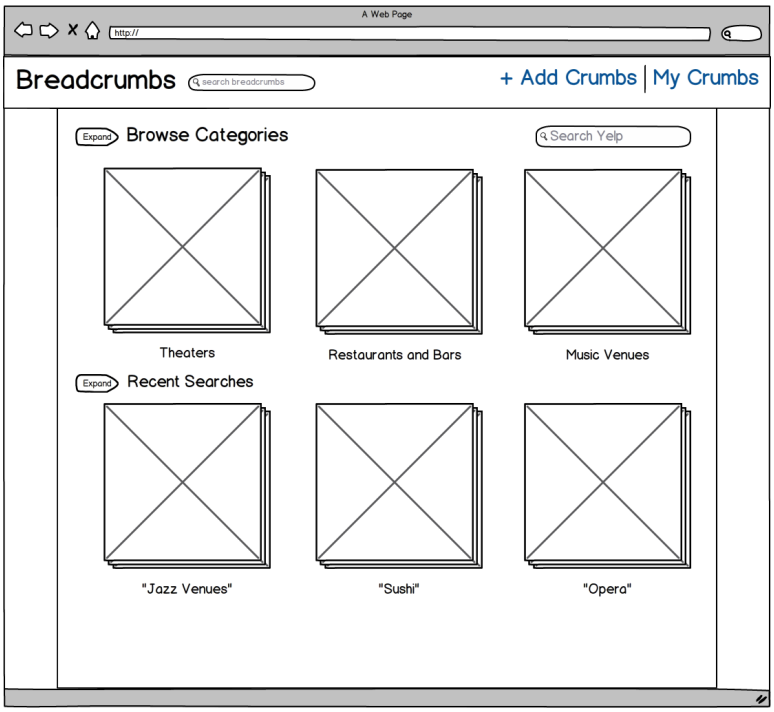


Browse View, or Add Crumbs

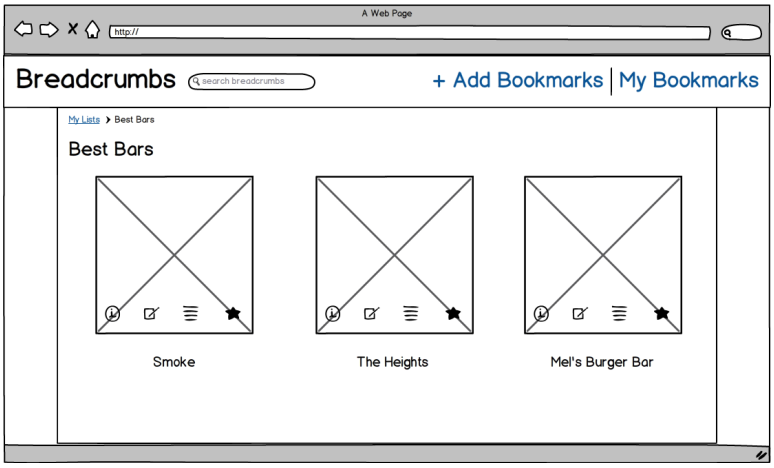
This is the initial markup of the browse view, accessed by clicking “Add Bookmarks”). It does not show the toolbar that appears when you click the box labeled “Search Yelp”.



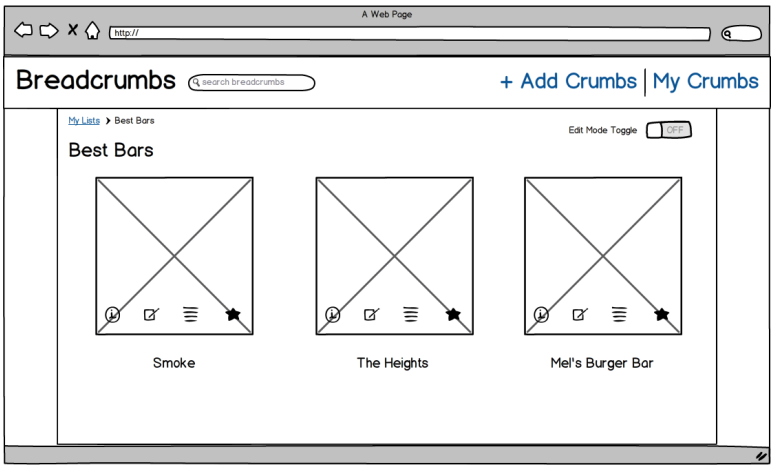
This is the final version of the browse view. As you can see, we added “expand” arrows to the “Browse Categories” and “Recent Searches” sections, because we wanted a way to show more results than pictured here. Horizontal scrolling was also discussed as an option, but it was ultimately vetoed because of some group members’ dislike for the feature.



Single List View



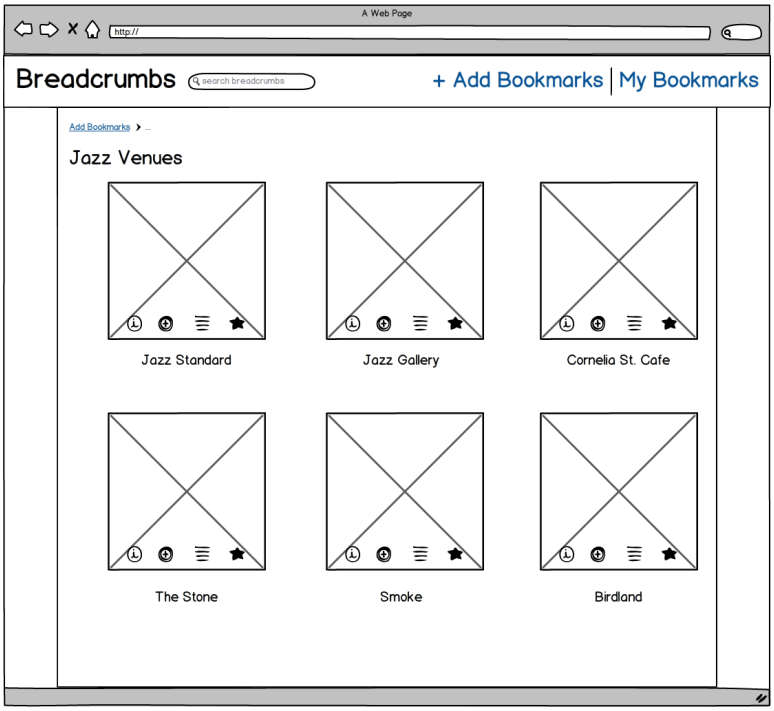
This is the initial markup of the single list view, showing a list called “Best Bars”. Each list can be accessed from the “My Bookmarks” page. The four icons over each picture have the following meanings: info, notes, list membership, and reviews. By hovering the mouse cursor over any of these icons, the app will replace the picture of the business with the corresponding information. Furthermore, by clicking on any of the icons, the selected information will freeze in place of the picture.



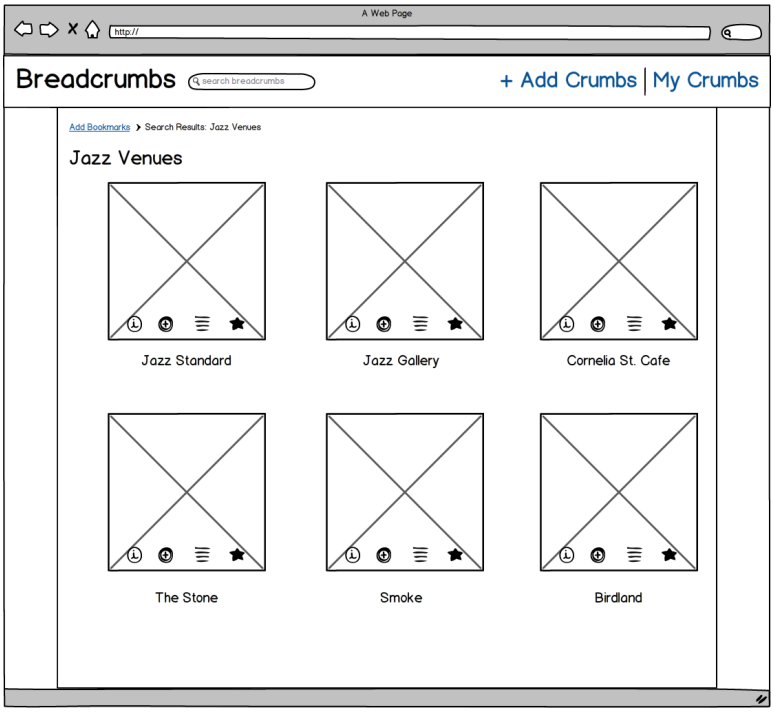
This is the final markup for the single list view. The only change we made was a toggle that switches the list in and out of “edit mode”. The “edit mode” feature allows the user to reorder and delete crumbs from the list.

Search Results View

This is the page that appears whenever you search for something in the “Add Bookmarks” page. This markup shows the results for the query “Jazz Venues”. The four icons here are info, add bookmark, list membership, and reviews. (The same as in the single list view, but with “add bookmark” in place of notes.) The hovering and clicking features are the same in this page as they are in the single list view page, creating consistency across our app.

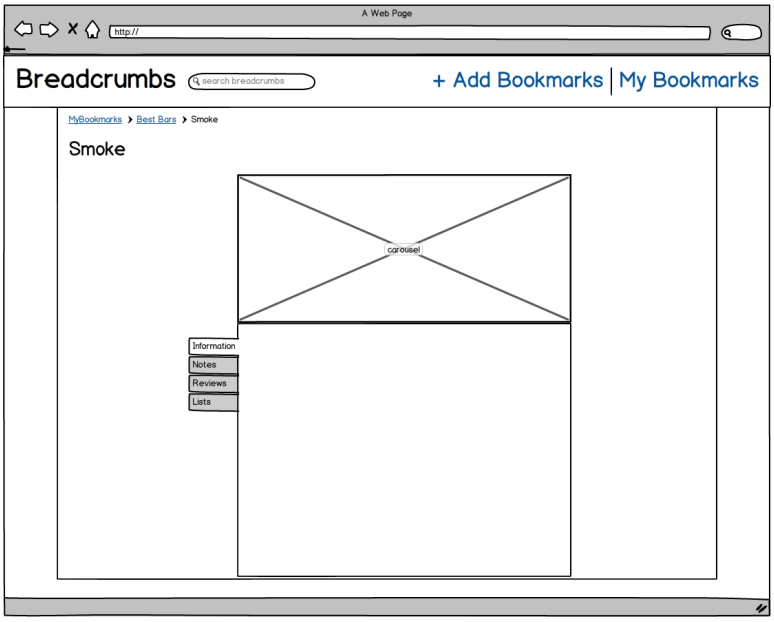


The final markup here is almost identical to the initial one. The only change is that we fixed the site map functionality at the top of the page. (Coincidentally, this site map feature is itself called “breadcrumbs”—no relation to our app.) This clearer site map helps our app more closely adhere to the third usability heuristic: user control and freedom. The user knows exactly how she got to this page in the app, and can easily return to the previous page if she found she made a mistake.

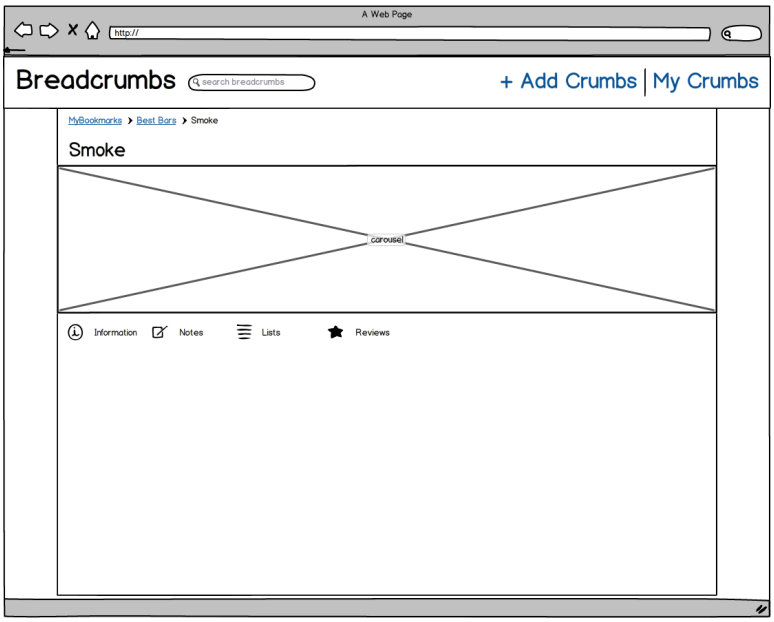


Single Item View, or Crumb View

This is the initial markup for the page that appears when you click on a crumb in one of your lists. As you can see, it contains tabs that correspond to each icon in the list view. It also contains a larger picture, and more space to read about the information, notes, lists, and reviews.



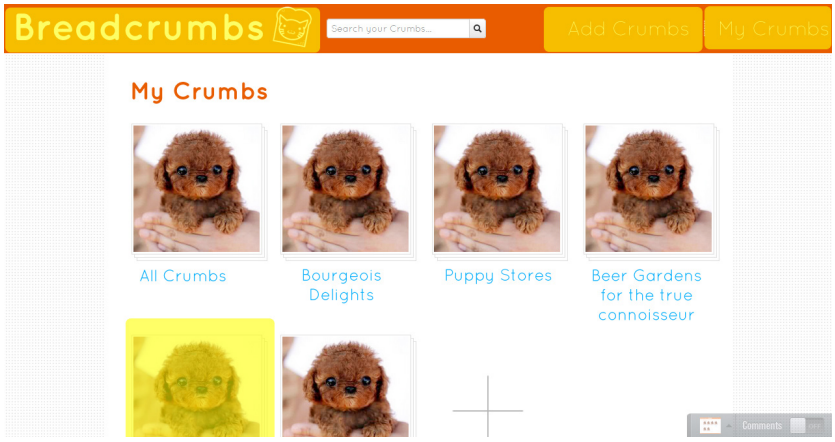
This is the final markup for the single item view. We increased the size of the picture, and also decided to use icons instead of tabs in order to switch between the information, notes, lists, and reviews sections. This decision made our app more consistent, as now every function is mapped with the same icon throughout every page in our app. This follows the fourth usability heuristic: consistency and standards. Each icon in each page does exactly the same thing as identical icons on different pages, and there is exactly one icon to represent each Breadcrumbs function.



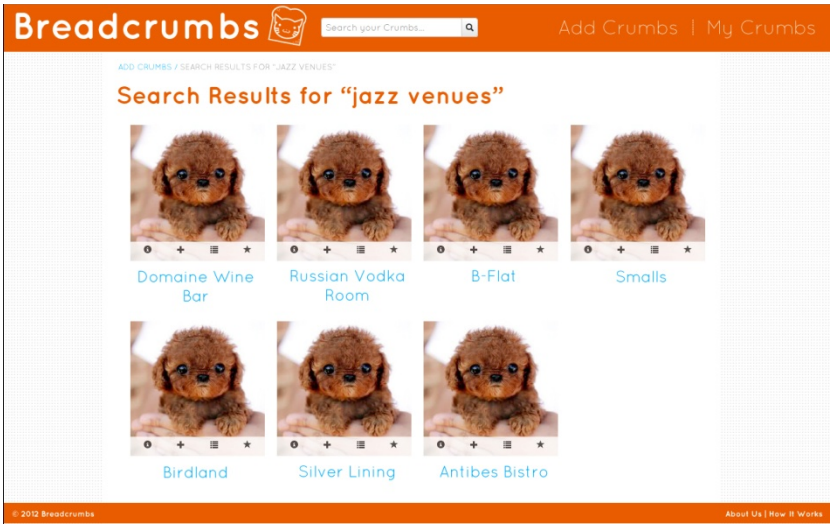
Testing Process

While we wanted to test our app both inside and outside the group, the extended amount of functionality required in the short timespan we were given for the project made it extremely difficult to design and finish the web application once, let alone creating the app, testing it, and then iterating it once. Our solution, therefore, was to use another app called InVision, an app that enabled us to create interactive prototypes with image files, for high-level prototyping after myBalsamiq. The project can still be accessed with the criteria below:

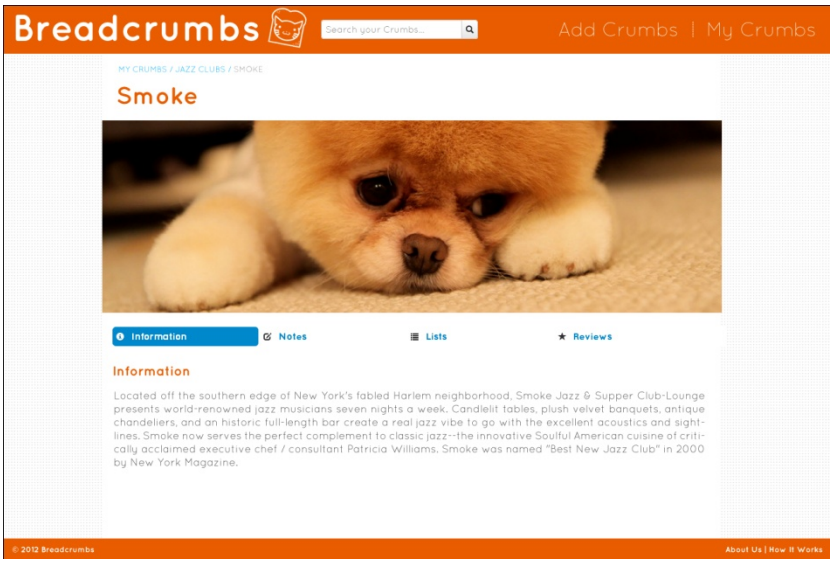
InVision Project: http://invis.io/X69N3KJZ
InVision Password: catintoast



InVision provided an invaluable resource for us during our internal testing and final prototyping process. As can be seen in the screenshot above, InVision allowed us to put in "hotspot" areas that would take a user to a new screen when clicked.



This is the final design screen for the search results page as implemented in InVision.



This is the final design screen for the single item page as implemented in InVision.

Each member of the team went through the InVision project and offered comments on the interactions that they thought were missing or could be improved. Thanks to our ability to do high-level, interactive testing in InVision that functioned similarly to a work app, we were able to discover a number of gaps in our design. For example, what screens were not present in our initial designs, such as the small interactions for adding notes and lists that we would expect on each page? What image would we use to represent an empty list, or how would we implement them in a logical, streamlined way in our app? For this, we were not able to iterate again in the design process, but these became our top questions when we began the implementation process. Adding notes become a function that a user could easily do while browsing through their bookmark lists; lists could be checked or not checked in a similar interaction.

We also noticed that although the application was consistent and easy for us to follow, it would be important to give a user constant feedback in a consistent way as they used the app. For this, we decided to rename the app "Breadcrumbs," like the breadcrumbs of Hansel and Gretel to find the trails they'd walked. This became a familiar analogy that any user would be able to understand. We also created a cat logo and visual identity to use throughout our application. The visual identity was created with a consistent color palette and a friendly typeface, Quicksand, to help forge a user's expectations of the app and the simplicity that it promised in bookmarking organization. In line with our aesthetic decisions, we decided to also make our language consistent with our app concept and call our bookmarks "crumbs."

Another interesting question was, what happens to the "Add Bookmark" icon when we actually add a bookmark—does it turn into a checkmark, grey out, or go

away? Here we decided to turn the icon into a checkmark, inspired by similar "thumbs up" and "thumbs down" icons in Facebook and Bwog. (Bwog is the website for Columbia's publication *The Blue and White*.) Other smaller issues came up as well, and our group would address them either in person, when we were working in the same room, or by email, when we were working remotely.

Software Engineering

At the very beginning of the project, Jonathan set up a number of development tools to help us keep our team focused and organized, such as Workflowy for task delegation, GitHub for version control, and CoffeeScript, Backbone, Underscore, and Mustache for augmenting JavaScript. Yanyi also introduced us to a testing and design application called InVision.

- Workflowy
    - Workflowy is the only non-software related item on this list, but it was so important to our process it is worth mentioning. Workflowy is an application that allows users to maintain and develop shared to-do lists. We used to take notes from meetings, delegate tasks, and to keep track of what the other members were doing.
- GitHub
    - Our group used GitHub as our version control system. Some of us were new to GitHub, but it ultimately proved crucial for us to keep track of project updates and to monitor our progress.
- CoffeeScript
    - CoffeeScript is a language that compiles directly to JavaScript, but has simpler and more intuitive syntax. It was much easier for us to code with CoffeeScript than JavaScript for this reason, and although there was a bit of a learning curve, it ultimately boosted our productivity.
- Backbone.js
    - Backbone is a JavaScript library that provided us with some of the most fundamental features of our app: models (for businesses), collections (for lists), and views (for application views).
- Underscore.js
    - Underscore.js is a library for JavaScript that provides a lot of functions that are built into other languages that have a stronger object oriented paradigm. The functions a split into categories: Chaining, Utility, Objects, Functions, Arrays, and Collections. A few examples actual functions are map, select, invoke, deep equality testing, and templating.
- Mustache
    - Mustache is a way to create dynamic templates for text in many different languages. This is particularly useful when working with html that needs to be changed based on backend operations. You can insert {{name}} tags into your code that will then take on the appropriate value based on the corresponding value in a hash. We use this to populate the image tags and text of businesses.
- InVision
    - InVision is a tool similar to myBalsamiq but with greater functionality. It allows for you to add interactivity with a focus on overlaying on top of images of designs rather than making wireframe outlines.