HW3 Report

https://eatbettereatboston.netlify.app

Data Description

For this project, we chose the dataset containing yelp reviews from various restaurants around the city of Boston. The exploration needs of our user include finding a suitable restaurant to visit somewhere in Boston while location and type of cuisine are not predetermined. It is also assumed that the user does not have a price range in mind as the yelp data did not provide us with the cost of each restaurant. The user will want to explore restaurants of various cuisine types, as well as their associated rating in order to make informed decisions regarding going out to eat in Boston.

To begin, we pre-processed the dataset a little bit in order to fulfill the exploration needs of our target users. First, we decided to remove any of the data points that did not contain a specific category – this field was left as null in the dataset. We thought it was best to remove these since users were coming in with no cuisine category preference at all, and this is exactly the field they wish to explore. Secondly, we also removed the data points that fell in the neighborhoods technically outside the city of Boston, since they appeared to be floating in space around the map outline, which did not make sense. These preprocessing steps took the number of datapoints from 344 to 254 restaurants.

Storyboards

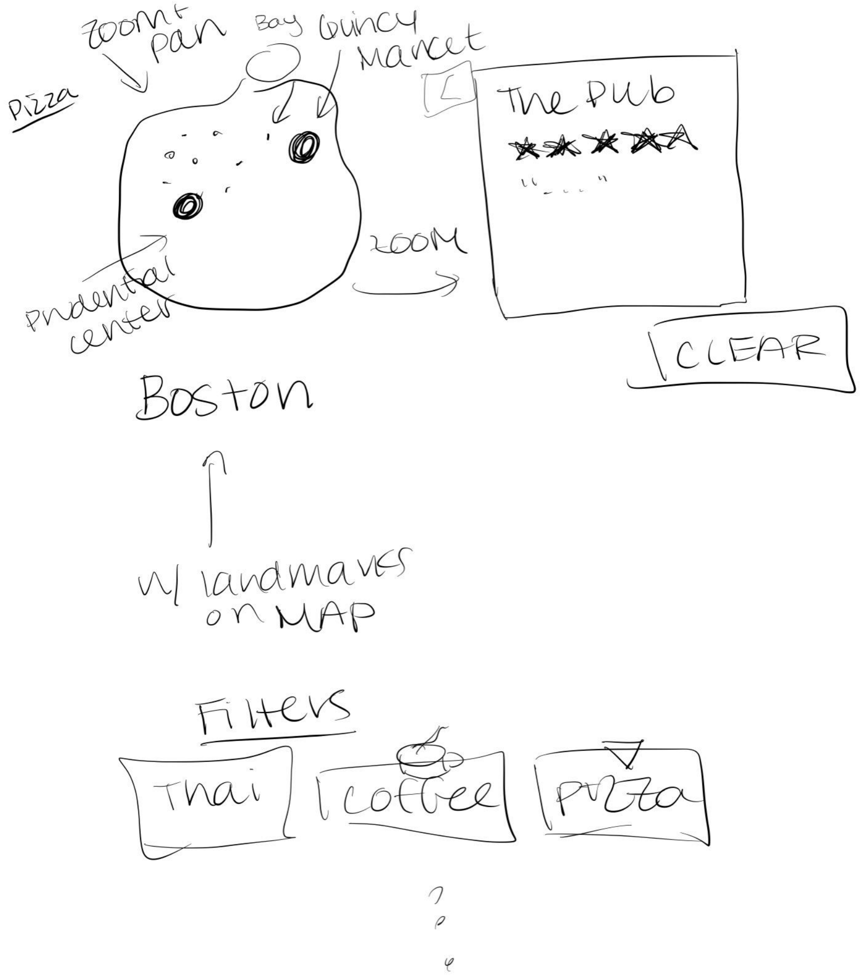
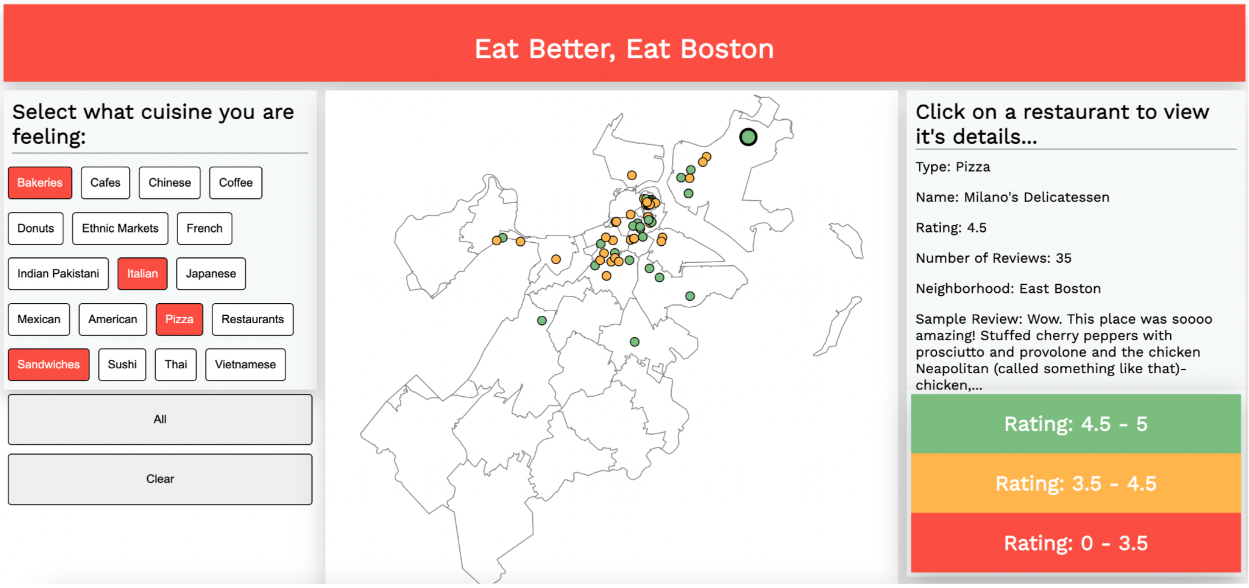


Figure 1: Original Brainstorming Storyboard Sketch

Figure 1 shows our original brainstorming session and the interactions associated with our preliminary sketch. We decided to use filtering by category as the primary method for filtering the data presented. Since users come in with no specific preference to the type of cuisine, we thought it would be helpful for the user to be presented with all of the various categories available, and then be able to select one, multiple, or all of them to show on the map visualization. Another interaction that we wanted to incorporate was panning + zooming on the map. Since at first glance the data points are packed densely in specific locations, it is hard to differentiate between the various restaurants on the map. That being said, zooming will allow the user to look more closely into a specific location and see each restaurant on its own without the overlap. Panning allows the user to remain zoomed in, but move to other parts of the map. This functionality is critical for being able to select specific restaurants accurately and being able to see the finer details of the map. Finally, the last interaction we designed was the mouseover interaction. When no restaurant (dot on the map) is selected, there is no further information about each restaurant besides its physical location on the map. However, when the user hovers over a specific restaurant, we want them to be provided with finer details. Thus, we designed a side panel that would populate all of the information about a specific location once the user selected some input.

Visualization



Eat Better, Eat Boston

Select what cuisine you are feeling:

Click on a restaurant to view it's details...

Type: Pizza

Name: Milano's Delicatessen

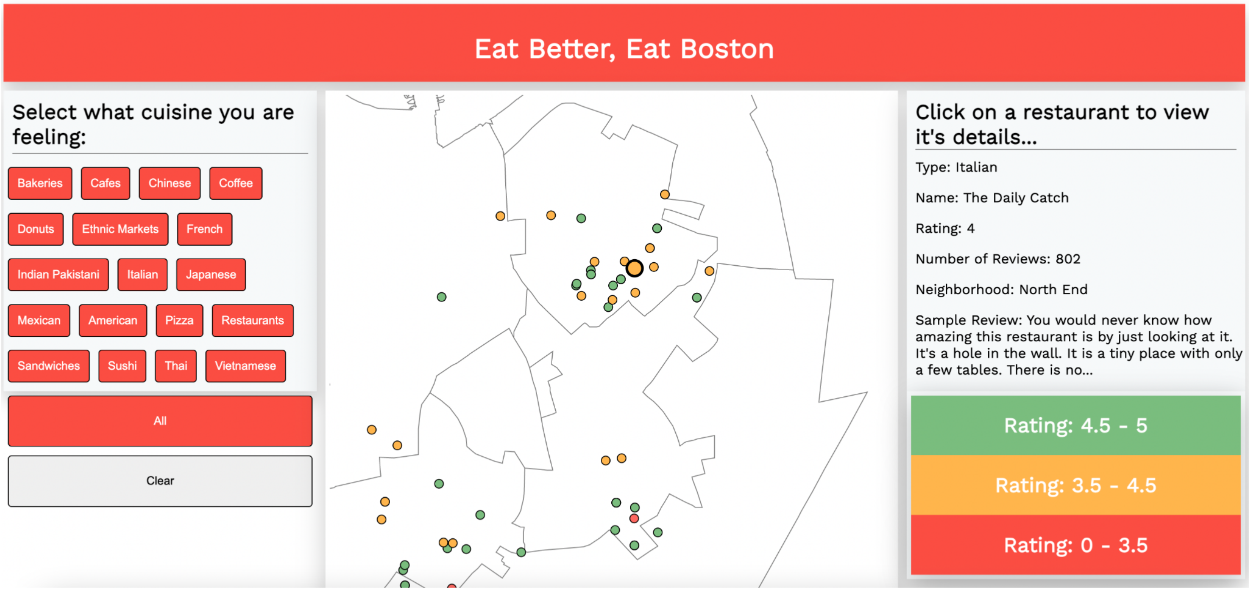
Rating: 4.5

Number of Reviews: 35

Neighborhood: East Boston

Sample Review: Wow. This place was soooo amazing! Stuffed cherry peppers with prosciutto and provolone and the chicken Neapolitan (called something like that)- chicken,...

Figure 2: Zoomed out map with selected cuisine types and specific restaurant information



Eat Better, Eat Boston

Select what cuisine you are feeling:

Click on a restaurant to view it's details...

Type: Italian

Name: The Daily Catch

Rating: 4

Number of Reviews: 802

Neighborhood: North End

Sample Review: You would never know how amazing this restaurant is by just looking at it. It's a hole in the wall. It is a tiny place with only a few tables. There is no...

Figure 3: Zoomed in map with all cuisines selected and specific restaurant information

Our application, “Eat Better, Eat Boston” allows the user to explore around Boston, finding restaurants of various cuisines and ratings. The user can select what type or types of cuisine they are looking for and then can immediately see the restaurants in Boston plotted on the map before them. Each restaurant is represented by a dot on the map, with the color of the dot corresponding to the quality of the restaurant's rating. The user can see where the restaurants are located in Boston as well as zoom into the map in order to see the areas that are more densely populated with restaurants in finer detail. By hovering over each restaurant's dot, the user can quickly inspect the type of cuisine, name, neighborhood, rating, number of reviews, and a sample yelp review of the restaurant. The user is able to see each restaurant in finer detail (details on demand!) as they parse through the map.

Trade-offs:

1. One tradeoff we considered was the difference between hovering and selecting a given restaurant's circle. While the click selection of a circle would allow the user to temporarily store the details inside the inspector to perhaps copy down, we thought the hover would allow for quicker and smoother exploration as the main goal of this visualization is to allow the user to explore restaurants all across Boston. The click selection method could potentially slow down this process, and if the user does need to copy down the information, they can simply remain hovering on a given dot. In addition, the mouseover effect increases the size of the circle and the boldness of the circle's border, making it extremely easy for users to tell which restaurant they have selected, as well as keep their cursor on the point to read the information in full.
2. Another issue that we considered was adding landmarks to the map. These landmarks would include popular places in Boston such as Faneuil Hall, the Commons, Fenway Park, etc. We included this idea in our original storyboard design plan. However, while we were implementing the design, we thought it might not be as helpful to users as we had at first intended. For each landmark, we placed a gold star on the map, and were going to label them by name. Unfortunately, the landmarks were hard to see with all of the other dots on the map, and the labels would simply clutter up the space. In addition, the gold stars might be confused with signaling very highly rated restaurants instead of landmarks. Ultimately, we decided it was best to stick with just the restaurants on the map and let the user discover the neighborhood data for every restaurant that should help orient them.
3. The last tradeoff/issue that we debated was having the filters and finer details panel occupy the same space, and having them toggle on and off depending if there was a point selected or not. Our original design included this idea, with the filters being present at the very beginning. As the user explores the map, if they end up hovering over a point, the filters panel gets replaced by the finer details panel to show specific details of that restaurant. When we thought about it more, however, we realized that this constant interchanging of the two panels would be very confusing and overwhelming for the user. Additionally, it is helpful for the user to be able to view the filter panel at all times to

know exactly which filters are being applied at all times. Thus, our final design implemented both panels on each side of the screen with the map located in the middle.

Development Process

Once our storyboard was complete, we began developing. Danny started by laying out the general structure of the page including a header, left sidebar for filters, and the rest of the page for the map. This initial idea was changed, however, to include a second sidebar on the right side of the page to house the inspector of each restaurant. We made this change because as mentioned above, initially upon hover, the filter section would become the inspector section. This would have been confusing to the user and so we added the inspector to the right side of the map. This gives the overall layout a symmetrical feel and guides the user to progress through the three stages of our visualization quite nicely left to right: start with basic filtering, then view the overall map, then view specifics of a given restaurant. Once we had the overall layout, Danny added in the filters, including “select all” and “clear all” buttons to help the user easily and quickly filter and undo their filtering. Next, Isabelle created the map and added the zoom feature. This zoom feature ended up being crucial to the visualization because it allows the user to penetrate into densely dotted areas and see the individual restaurants that reside very close to one another. This also allows the user to hone in on a particular area/neighborhood of interest. The zoom also is incorporated with the user’s scroll so it is both familiar and easy to adjust or correct a mistake made when zooming. Once the map and zoom was complete, Isabelle wired up the filters to the restaurant dots and added in the hoverability. Then, we coded in the dynamic inspector and the color legend for rating. We decided to leave the fields (type, name, rating, etc) blank but still visible when the user is not hovering over a specific restaurant. This is because we thought it was important for the user to see what information is available to them before they even select any restaurant. Finally, we added some basic text atop the filter panel and inspector panel to slightly nudge the user with what they should be doing. This differs from our initial design, but the text serves as a very primitive step by step guide that gives the user the different tasks they can complete with our tool. Select the foods you are interested in, look around the map of Boston, and get details on a specific restaurant of their choosing. In this way the user knows exactly how to use the tool, and there is less ambiguity than if the visualization was just thrown at the user with zero explanations.

Work Breakdown

For this project, we aimed to keep the workload at a 50/50 balance between the two of us. Initially, we both participated in a brainstorming/storyboarding session in order to flesh out our ideas for the visualization. Once we were happy with our initial design and associated interactions, we divided up the coding work. Danny was tasked with creating the filters and overall layout of the page, while Isabelle was responsible for the map. We looked over one another’s code and worked through any major bugs together. Finally, we divided up the report into sections for each person. Overall, the coding took the most time, more specifically the map visualization. We ran into issues with the zooming and the filtering of data points using a data join, but were ultimately able to get them resolved.

- Isabelle: brainstorming session (1 hr), code – map visualization/code review/bugs (7 hrs), report (2 hrs)
- Daniel: brainstorming session (1 hr), code – layout/styling, filters, code review/bugs (7 hrs), report (2 hrs)