

## **Salt Lake City Food Inspection Visualization**

Ishrat Jahan Eliza, u1472593, [u1472593@utah.edu](mailto:u1472593@utah.edu), Arleth Salinas, u1472543, [u1472543@utah.edu](mailto:u1472543@utah.edu), Nathan Gorski, u1472279, [u1472279@utah.edu](mailto:u1472279@utah.edu)

GitHub Repository: <https://github.com/dataviscourse2023/final-project-slcfov>

### **Background and Motivation.**

We will be creating an interactive visualization of food inspection data from the Salt Lake County Health Department. This visualization will be helpful as it informs possible restaurant patrons about the overall cleanliness of whatever establishments they may be seeking to dine at. We have also found the information in these data humorous, in a sardonic way. For example, we saw a sushi burrito restaurant that received over 70 health code violations in the past four years. Additionally, one of our group members is particularly interested in urban geography, and this project has an urban geological component from the spatial context of the establishment locations. Furthermore, we determined that this dataset was accessible to us because the Salt Lake City Health Department provides thorough definitions for the attributes of these data.

### **Project Objectives.**

The primary questions that we are trying to answer are as follows:

*Which local restaurants and restaurant chains in Salt Lake County are the cleanest and least clean? Which health code violations are the most and least common? How has restaurant compliance with the health code changed over time?*

We aim to provide a visualization to help potential restaurant patrons make informed decisions before they dine. The benefits to this are immediate: such a visualization will help customers make informed decisions about where to dine in terms of any restaurant's sanitation. Additionally, one can see whether a restaurant has improved or degraded in health code compliance over time. Finally, visualizing the data collected by the Salt Lake County Health Department makes this information more accessible to the public.

### **Data.**

The data was obtained directly from the Salt Lake County Health Department. We made a phone call to the county, and they emailed us a CSV file containing the results of every health inspection since 2019. The data is comprehensive; it includes every restaurant, its address, and any (if any) violation. To that end, we do not need much more data for our visualization. For violations, both the code number and title are listed. The code numbers are grouped into code families (e.g., 4.2.14) and are in families 4.1 through 4.9. The titles of these families of codes will be manually entered from the Salt Lake County health code, which can be found here:

<https://slco.org/globalassets/1-site-files/health/regs/food.pdf>

## **Data Processing.**

### *Data Cleaning*

First, the names of restaurants need to be listed in a standardized way. For example, one restaurant has an address “565 E 2100” in its name column “565”. It is unclear if this is the actual name of the restaurant. We need to identify data points whose names and/or addresses are unclear and retrieve the correct names. We plan on doing this by checking Google and Yelp. Moreover, chain restaurants often have a number after the name (e.g., “Subway #10757”). We plan on deleting these numbers for readability. Note that different locations of the same restaurant may be identified by their address.

This data set also includes cafeterias at elementary schools and retirement homes. We have decided to remove public schools and residential facilities to ensure only publicly available facilities are represented. Finally, some codes are not considered violations but rather notifications. For example, “cleaning education provided” is not a code violation. Thus, we will remove such points from the data.

### *Data Analysis*

There are various quantities we plan on deriving from the data. One of these derived attributes is whether a given violation is critical or non-critical. This is technically implemented in the current data as an asterisk following the health codes violated. However, it will be far easier for our purposes to create a new column indicating whether or not the health violation is critical. After this post-processing, we will remove the asterisks from the codes.

Another attribute we want to derive is the town that each establishment is located in. The data only lists each establishment's street address since addresses are unique in Salt Lake County. We will obtain this information for each establishment by combining two APIs. One API is through Salt Lake County, allowing us to send an address and receive the corresponding coordinates. Another API is through OpenStreetMap, which allows us to enter coordinates and receive the town located at those coordinates.

Finally, we will also calculate the average number of critical and non-critical violations per inspection for any restaurant.

### *Additional Implementation Notes*

Our data comes in the form of a comma-separated file. It is small enough to work within Excel. However, if we find that Excel is insufficient for our needs, we plan on using the Pandas Python library to load our data and conduct our data processing.

For standardizing the names of the establishments, we plan on manually checking the restaurant names in the sheet and then editing the names accordingly. Fortunately, the number of restaurants in Salt Lake County is reasonable to inspect manually. We will manually verify which restaurants are not publicly available and delete those rows.

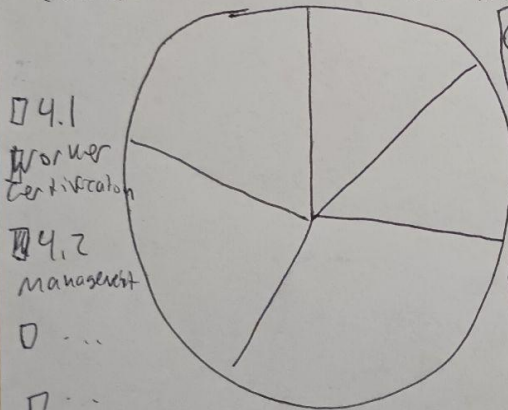
### **Visualization Design.**

Found below, the first two photos are Nathan's prototype (Figure 1), the next two are Eliza's prototype (Figure 2), and the next two are Arleth's prototype (Figure 3). The final two photos are our final designs (Figure 4). Figure 4(a) corresponds to the must-have features we thought to implement first, and Figure 4(b) corresponds to the optional features we agreed upon. Our prototypes were based on the features we decided upon, and we conceived of our designs based on these features. The agreed-upon features to consider were a menu containing the establishments, visualizations containing aggregate statistics of the inspection data, and visualizations of individual establishment statistics.

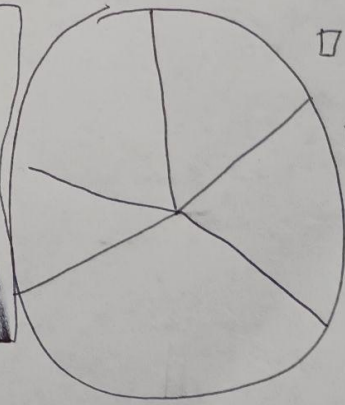
# Minimal Implementation



Common codes for Sushi Burrito:



Click on a slice on the left pie chart to change the right pie chart. This text wouldn't be in the display.



other

4.1 Worker Certification	4.2 management	4.2.15 Tobacco	4.2.12 hand washing
4.1.1	4.2.1	4.2.15.1	4.2.12.1
4.1.2	4.2.2	4.2.15.2	4.2.12.2

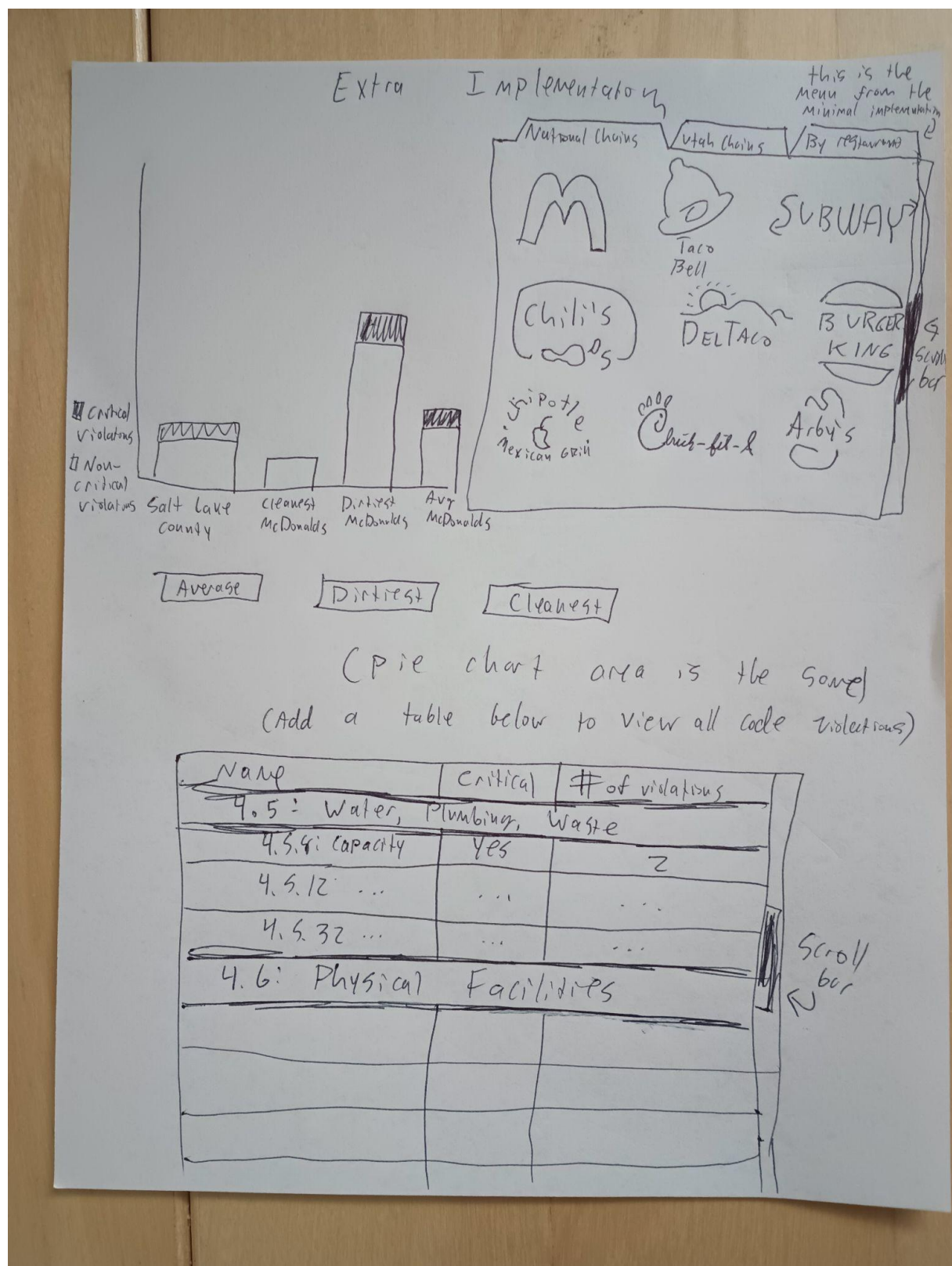


Figure 1: Nathan's Prototype 1 & 2



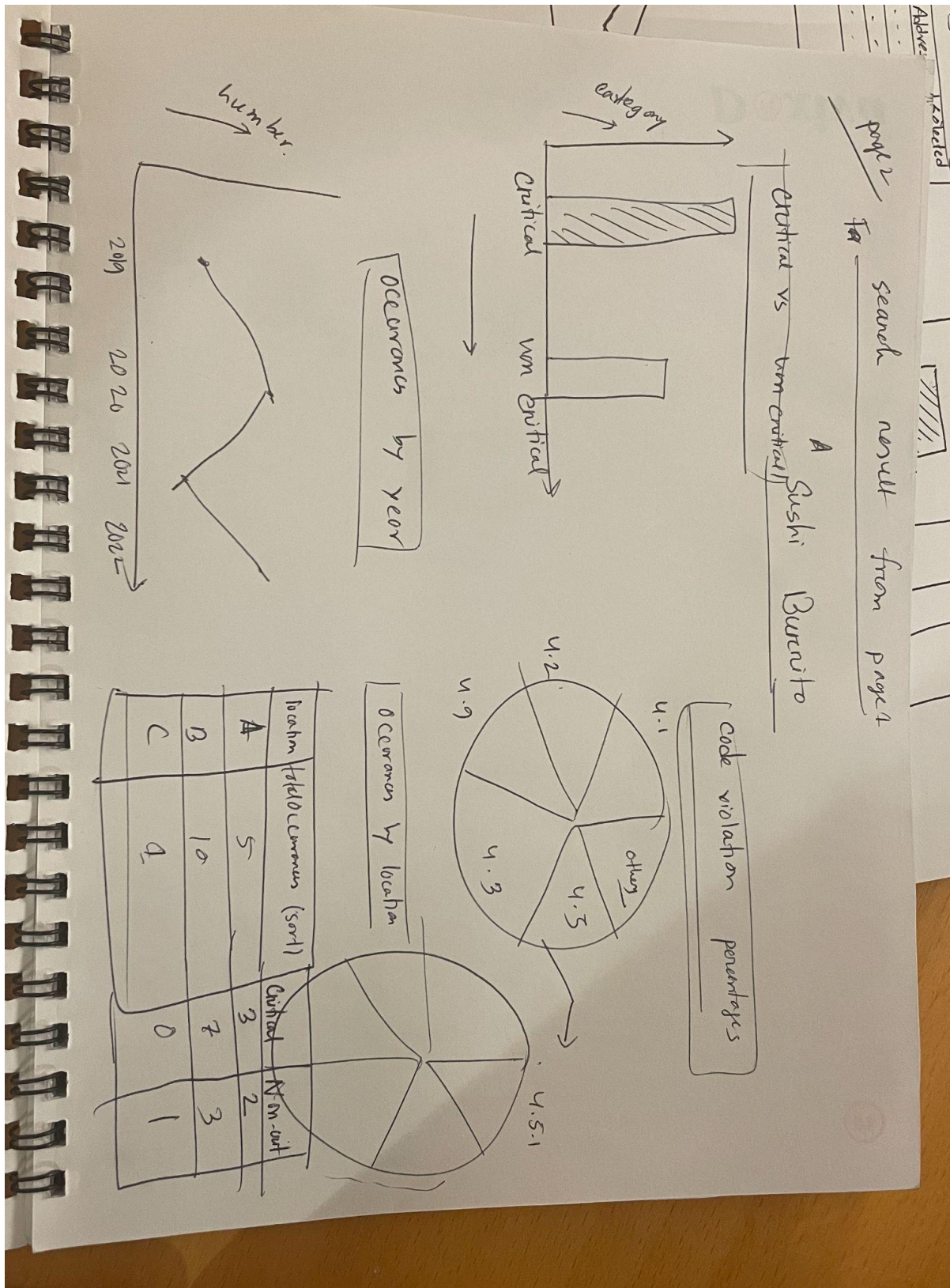


Figure 2: Eliza's Prototype

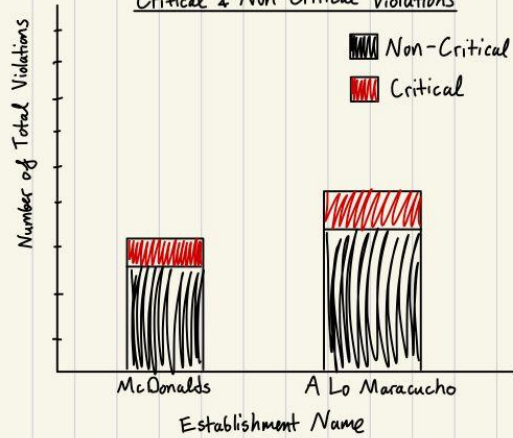
# SLCFIV Salt Lake City Food Inspection Visualization

Menu  
↳

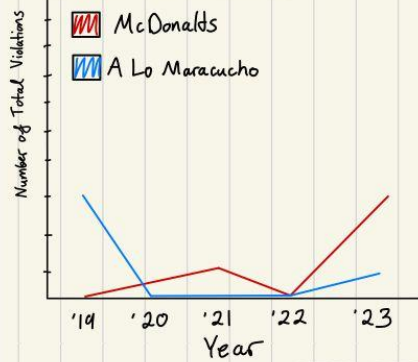
Search  Map Violation Dictionary

Establishment ID	Name	Address	Selected
1	McDonalds		<input checked="" type="checkbox"/>
2	A Lo Maracucho		<input checked="" type="checkbox"/>
3			<input type="checkbox"/>
4			<input type="checkbox"/>
5			<input type="checkbox"/>
6			<input type="checkbox"/>
7			<input type="checkbox"/>
8			<input type="checkbox"/>
9			<input type="checkbox"/>
10			<input type="checkbox"/>
11			<input type="checkbox"/>
12			<input type="checkbox"/>
13			<input type="checkbox"/>
14			<input type="checkbox"/>
15			<input type="checkbox"/>
16			<input type="checkbox"/>
17			<input type="checkbox"/>
18			<input type="checkbox"/>
19			<input type="checkbox"/>
20			<input type="checkbox"/>

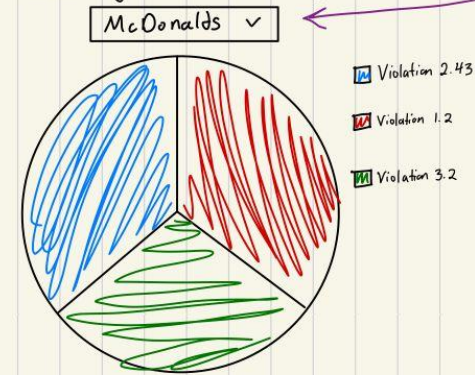
Number of All Critical & Non-Critical Violations



Violations Over Time



Number of Violation Code Types



Select Restaurant, currently either McDonalds or A Lo Maracucho

Custom Search

Aggregate View

# Custom Search-Menu Views

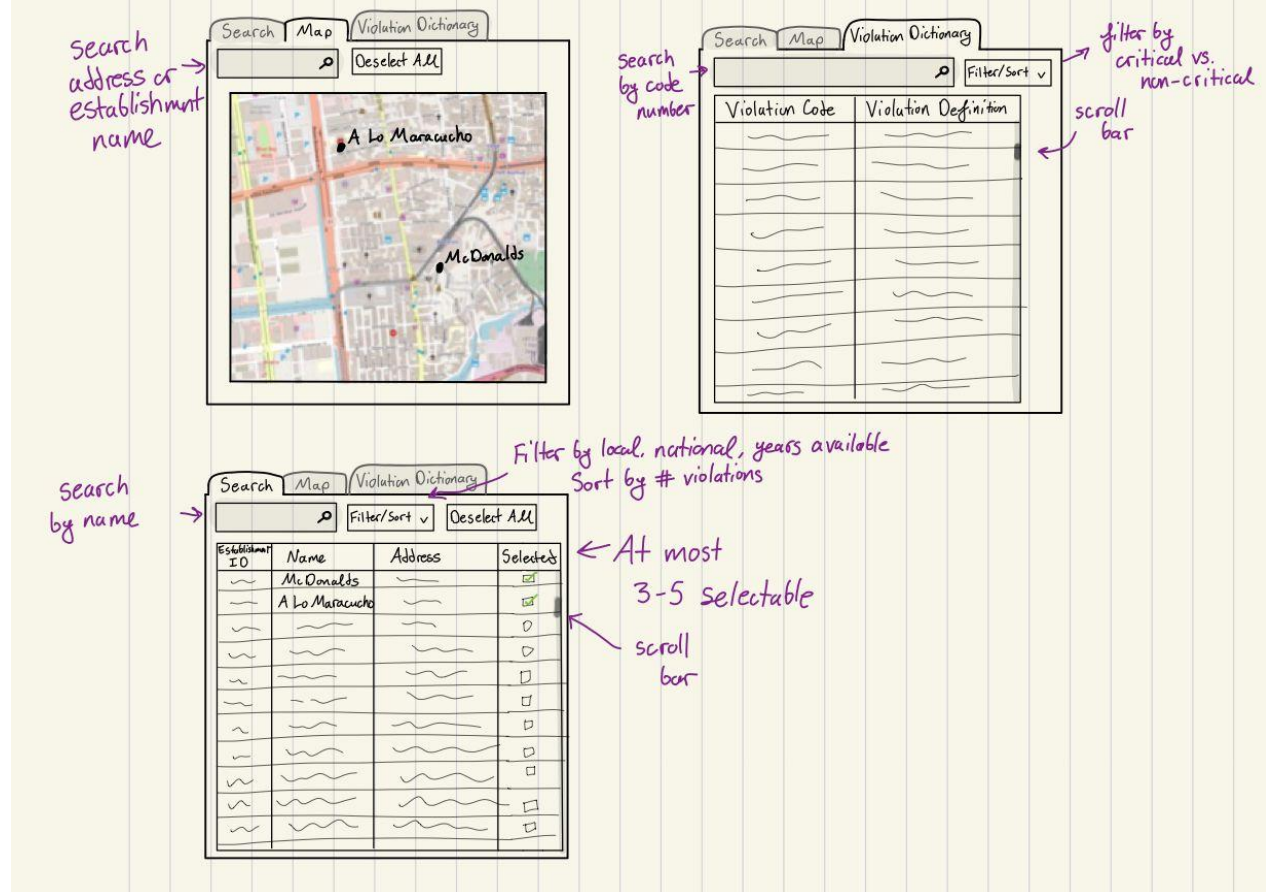


Figure 3: Arleth's Prototype 1 & 2



# SLCFTV

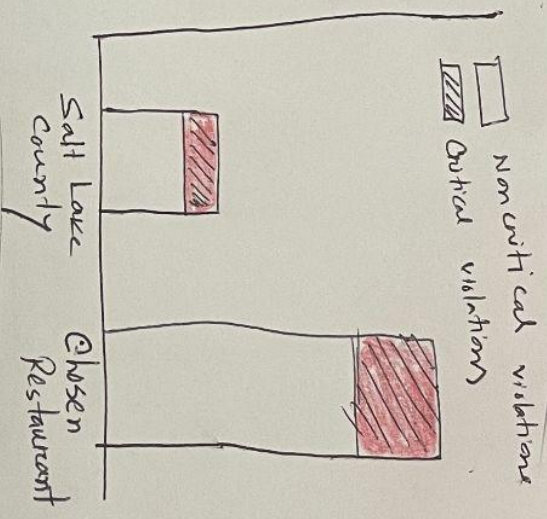
Search

☐ Filter ☐ Sort

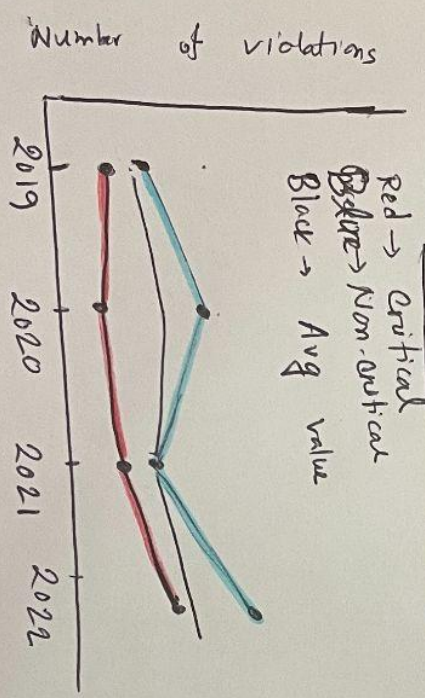
☐ A ☐ B ☐ C ☐ D ☐ E ☐ F ☐ G

Name	Address
McDonalds	
ALOM	

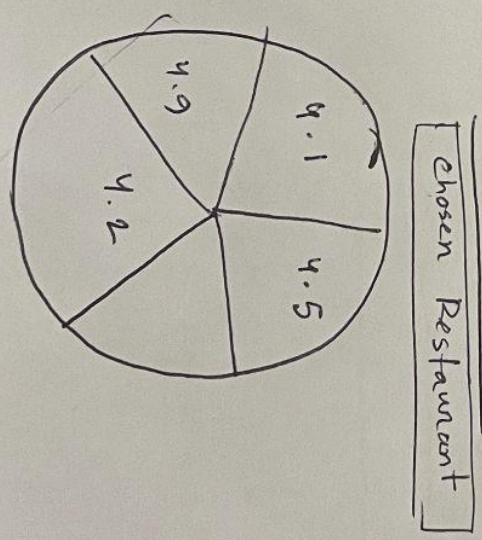
## Salt Lake City Food Inspection Visualizer



### Code violation progression



### Code violation types





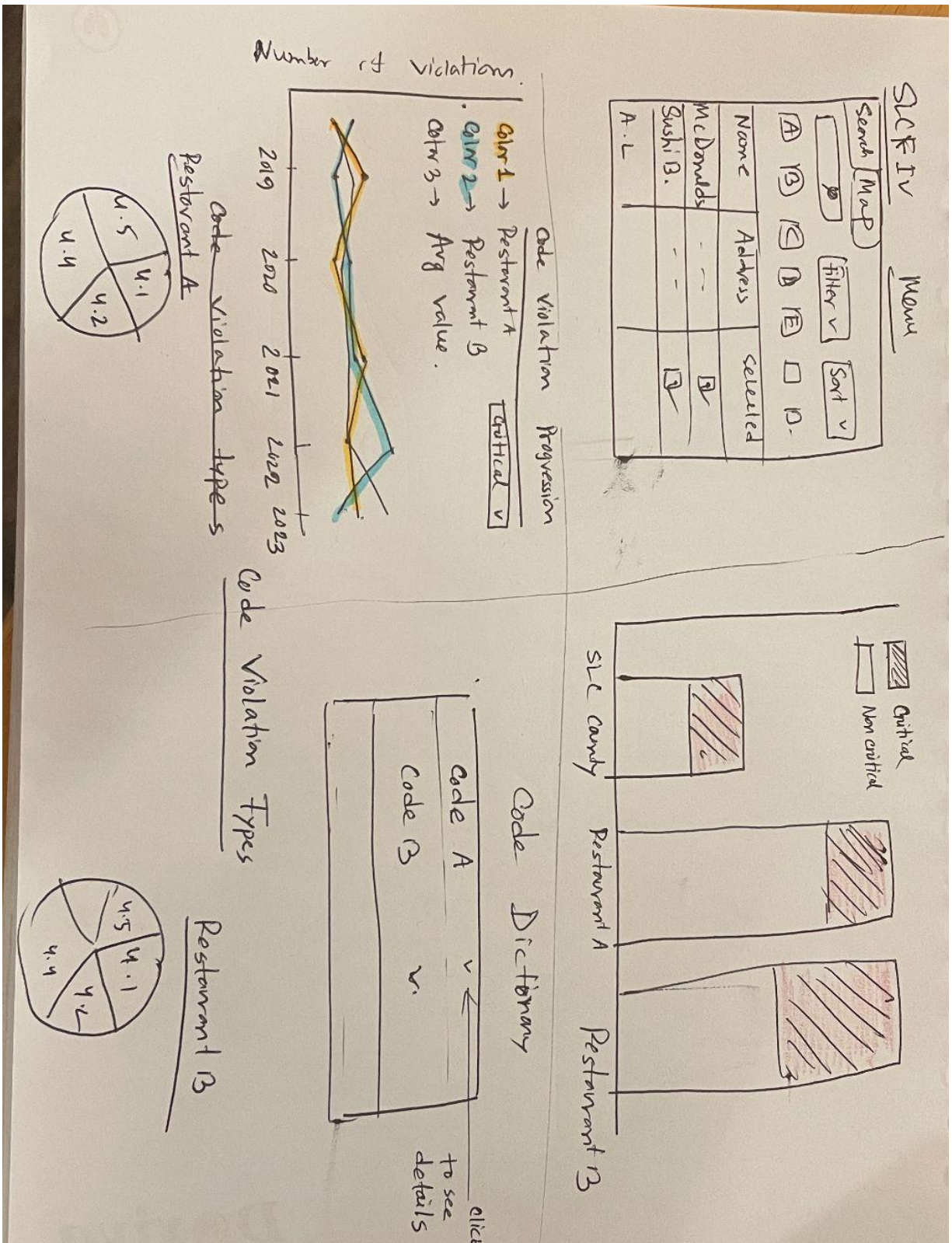


Figure 4: (a) The Must-Have Features, (b) The Optional Features

### **Must-Have Features.**

Our project should have a menu allowing you to select individual restaurants to view their code violations. This menu will allow you to search restaurants by name, city, or address. There will also be buttons corresponding to the different letters of the alphabet allowing you to cut down your search further. Additionally, this menu will have the option to sort your selection alphabetically or by number of violations.

Another must-have feature is a stacked bar chart. The stacked bars will have one section for the average number of non-critical violations per inspection and another for the average number of critical violations per inspection. There will be two bars, one corresponding to the restaurant that was selected and the other corresponding to the average values for Salt Lake County.

Additionally, we would like to have a line graph that displays each of the restaurant's inspections over time. There will be a dropdown menu at the top of the chart, allowing you to choose between displaying critical, non-critical, or total violations over time. Hovering over each vertex of the line graph will show the value at that point. The Salt Lake County average would also be displayed in the same way, where the average is taken over the whole calendar year.

Finally, we will have a pie chart which shows the breakdown of which code violations were present at the restaurant. Because there are so many possible codes, the codes would be broken down by code family (e.g. 4.1) rather than by code (e.g. 4.1.23).

### **Optional Features.**

One feature that would be nice to have would be the option to compare two restaurants side by side. This would involve a modification to the menu allowing for the selection of two different restaurants. This could take the form of two different tabs at the top of the menu allowing you to differentiate whether you are selecting restaurant 1 or restaurant 2. To compare both restaurants, the bar graph would have an extra bar and the line graph would have an extra line each corresponding to this new restaurant which was selected. A new pie chart would also be added showing the code breakdown for that restaurant.

One modification to the pie chart that would be good to have would be to show a breakdown of all of the codes corresponding to the restaurants' violations rather than just restaurant families. To implement this, we would allow the user to click on a slice of the pie chart corresponding to one code family in order to display a second pie chart below. This second pie chart will show the breakdown of code violations corresponding to the selected code family for that particular restaurant.

Another feature that would be good to have would be a dictionary of codes. This would allow you to obtain a more detailed description of the violation corresponding to them. This menu could give you the option to search first by code family, and then by specific code.

Additionally, we could have a map view that allows the user to locate the restaurants of interest. This would prove helpful for someone wanting to search for restaurants at specific locations, however, this provides convenience and not necessarily new insights.

### **Project Schedule.**

- Finish data processing by 9/22
- Functional menu by 9/29
  - In particular, the menu should contain all establishments, be searchable, sortable, and filterable.
- Bar chart by 10/6
  - At this point we should be familiar with using HTML, JavaScript, D3, and any other tools we identify.
- No work over fall break
- Line chart by 10/20
- Pie chart by 10/27
- Clean up menu. Must-haves completed by milestone on 11/3
  - Ensure proper menu functionality and communication with visualizations.
- Implement establishment comparisons by 11/10
  - We should be familiar with how our data structures and visualizations communicate at this point, thus allowing us to extend our visualizations with this comparison functionality.
- Implement the sub-level pie chart by 11/17
- Minimal work over thanksgiving
- Cleanup by submission deadline on 11/30