Appetize Visualization

Final Report for the Information Visualization Course (2019)

Amirhossein Abbasi
Electrical and Computer Engineering Department
The University of British Columbia
Vancouver, Canada
amirosein@ece.ubc.ca

Arya Rashtchian
Computer Science Department
The University of British Columbia
Vancouver, Canada
aryara@cs.ubc.ca

***Abstract*—We designed a visualization system comprised of different views to help restaurant owners with different tasks including designing a better menu, identifying loyal and non loyal customers and being able to manage their inventories in a more efficient and automatic way. We conducted a focus group with a number of restaurant owners to get their feedback about the designed views. Based on their positive feedback, we implemented a set of views that can be used in the web application which is designed for restaurant owners.**

Keywords—Information Visualization;

I. Introduction

Appetize is a Vancouver based startup that aims to streamline the process of dining in for its users. We provide two platforms: First is a mobile application that is designed for customers who are our end users, and the second one is a web application which is designed for restaurant owners. The mobile application lets customers take a look at the digitized menu, order their desired food, and pay for it. The advantage of this application is that customers can use it in a variety of restaurants instead of downloading an app for every single restaurant. Therefore, we are able to collect data that comes from the interaction of users and the mobile app, on top of that do data science, and give insightful information to restaurant owners via the web application. Besides, restaurant owners can define their food items and their menu in the web application. The goal for the project of this course is to design a visualization system that helps restaurant owners better design their restaurants, establish a better relationship with their customers, and compare their restaurants with similar restaurants.

II. Related Work

There have been some products that are currently being used by customers when they want to dine out. The most famous one is Google Maps, which lets users explore different restaurants on the map and see images and menus of each restaurant. Google Maps also shows feedback and reviews of all customers and also visualize crowdsourced data to show busy hours.

However, with Google Maps, you can not order through the app, so it does not have data related to customers' orders. Also, it lacks the process of keeping track of ingredients to help managing inventories. Thus, they do not offer any visualization for restaurant owners in this regard. In general, Google Maps does a good job providing insights for customers to choose their desired place to eat by showing feedback of others. In general, Google Maps is a great option to choose where to go, however, it is not as useful when you are at the restaurant. In addition, it does not provide any sort of visualizations for restaurant owners to assist them manage their restaurant in a better way.

Yelp is a more relevant product that targets helping both customers and restaurant owners at the same time. It is one of the most widely used restaurant information software across the United States. Although it visualizes the number of views for each restaurant and shows how customers end up reaching their restaurant (through ads, reviews, etc.), it does not provide restaurant owners with any sort of visualizations to manage their inventory. Also, it does not offer any visualizations to help restaurant owners find loyal customers and manage their promotions accordingly. It does not provide any choice for comparing your restaurant with the average of others by taking advantage of visualization tools.

There are some works for visualizing restaurant data in the research context too. Fukuhara has worked on improving service processes based on visualization of human-behavior in one chapter in his book [1]. This work tries to enhance the efficiency of indoor services in restaurants. Anil Bilgihan has analyzed online customer reviews for restaurants [2], as well as some other works in this area [3][4][5][6]. But generally speaking, researchers were more focused on the tourism and hotel industry, and there is little analysis directly for visualizing information for restaurants.

III. Data Abstraction

A. Domain-specific data:

To briefly describe what kind of data we have, we narrow them down into two categories.

The first category is all the detailed information about food items. The main goal of these data is to provide enough information about each particular food item. These data are related to menu, ingredients, categories of food items. Ingredients data are collected through entering their name and capacity directly.

The second category represents information about customers and their relationship with different restaurants.

B. *Abstracted Data:*

There are different types of data in the system. Below we show all related data schemas as well as the important columns and their data types.

- Restaurant: Represents restaurants in the system. It contains data to help us discriminate one restaurant from others. Besides, it provides some data like the location of a restaurant to use in the finding-similar-restaurants algorithm. This table has all the information for one restaurant like phone number, address, etc.
  - Name, phone number, address, and general information: String
  - Location: Present the longitude and latitude of a restaurant. [geographic]
  - Views: Number of views, time of view, and the user who viewed the restaurant through the app.
  - Menu: represents menus in the system. It contains different categories of food items.
  - Restaurant_ID, name: Name of the menu and the restaurant that the menu is for. [Integer] [String]
  - From_hour, duration: Shows the working hours of each menu [time data]
  - Views: shows how many times each menu has been viewed through the app [Integer]

- Category: represents a group of food items.
  - Name: The name of the category [String]
  - Items: a list contains indexes to food IDs that are assigned to this category [list]
  - Position: It shows where that category is placed within the menu. [Integer]
  - Item: represents food items in a restaurant.
  - Name, price, and the restaurant.
  - Views: It shows how many times the item has been viewed through the mobile app.
  - Ingredients: A list of indexes to relevant ingredients.
- Ingredients: Represents all the ingredients in the system. It has the data for how much of each ingredient is remained and what the maximum capacity of the container for each ingredient is.
  - Name, restaurant, and other general information
  - Remained and capacity: [Float]
- Order: Represents food orders placed by customers. It has the data for the food items in it as well as the data for the person who placed the order. Also, it has the data for the location and the time when the order was placed. In summary, for each order we have:
  - The user
  - Items and their quantities
  - Time of scanning the barcode on the table
  - Time of paying the order
  - Percentage of the tip

IV. Task Abstraction

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

A. *Domain-specific Tasks*

First, we provide a clear definition of the user in the system to list their requirements later. Users are people who are in charge of keeping track of how well the restaurant serves its customers while keeping the restaurant still profitable. Thus, users could be restaurant owners or the managers of restaurants.

In some cases, users are just members of the restaurant staff who make higher-level decisions. All possible target users of our visualization tool are people who are not necessarily experts in science or mathematics. Since users might come from different backgrounds, we aimed to build straightforward but at the same time, insightful interface and visualizations which are easy to use by users.

To define the tasks in a top-down order, we start with the final goals of the users. In most cases, the restaurant’s profitability is the users' top priority to use the analytics. There is no doubt that the success of a business depends on lots of other factors which might or might not be visible. Different factors such as how satisfied customers are with the food or how busy the restaurant is all have implicit or explicit effects on the success of a restaurant. Here we take into account some of the factors which are addressable via a visualization tool. The second important goal would be to help users manage their restaurants better. According to restaurant owners, managing internal tasks of the restaurant can be very painful, and a rich visualization system can be beneficial.

Below we provide a list to delve into user requirements a little more in detail.

- Insights into business profit.
- Insights into user satisfaction.
- Insights into the quality of their service
- Insights into helping restaurant owners manage their restaurant in a better way.

Although these factors are correlated to each other, they can help us classify tasks better.

B. *Abstracted Tasks*

- Restaurant owners want information about whether an item is profitable or not.
- Restaurant owners want to redesign the menu based on customers' feedback.
- Restaurant owners want to identify loyal/non-loyal customers to give some promotions.
- Restaurant owners want to know about the busy times of their restaurants to manage their resources in a better way.
- Restaurant owners want to compare their restaurants with similar restaurants (based on some defined similarity factors)
- Restaurant owners want to know how empty their food containers are, to effectively manage their inventories. (This is extremely useful for people who have multiple restaurants

(V) Algorithm

We have defined two algorithms in our system. One algorithm is to determine users' loyalty to each restaurant and the other algorithm to find similar restaurants to one particular restaurant. None of these algorithms are implemented and these are considered as future work.

(A) Customers' Loyalty Algorithm

The goal of this algorithm is to assign a number in the range of [0,1] to each customer to indicate loyalty towards one particular restaurant.

The idea for implementing this algorithm is to leverage a machine learning model that takes a set of labeled users and uses some features to guess the value of loyalty for new users.

We categorizes customers' visits into three classes based on how recent they are. Each class is considered as a feature in our model. The first step for training the model is to label a set of customers as loyal, semi-loyal, and not loyal and assign 1, 0.5, and 0, respectively, for their loyalty value. Then the algorithm fits a linear regression model using those three features as well as some other features. After training the model, it will be capable of approximating the loyalty values for unlabeled customers.

Unless we have real data, we can not evaluate this model. The model might be biased to what we have generated since the data is synthesized and might be unrealistic. Moreover, labeling the data requires to be done by getting help from some domain experts.

(B) Finding Similar Restaurants Algorithm

This algorithm gets one restaurant as input and outputs a list of restaurants that are considered to be similar to that specific restaurant based on some predefined criteria. These criteria are gathered mainly from a focus group that we conducted with restaurant owners and people who are experienced in the restaurant industry. The criteria are as follows:

- Neighborhood

One of the most critical factors for restaurant owners to attract more customers is to stay competitive in their own neighborhood. Finding similar restaurants in the same neighborhood is important for restaurant owners as it happens a lot for customers to decide between multiple restaurants in one specific area. We are using longitude and latitude to find all the restaurants that are within a radius to one specific restaurant.

- Price

The other factor that makes two restaurants similar is their price. This is also an important factor for people who want to choose a restaurant when they want to dine out. It is definitely useful to take the price into account when we are looking for similar restaurants as owners of fast-food restaurants do not want to compare their restaurants with high-end restaurants. In this regard, we calculate the average of all the food items and consider that number as the price for each restaurant.

- Environment

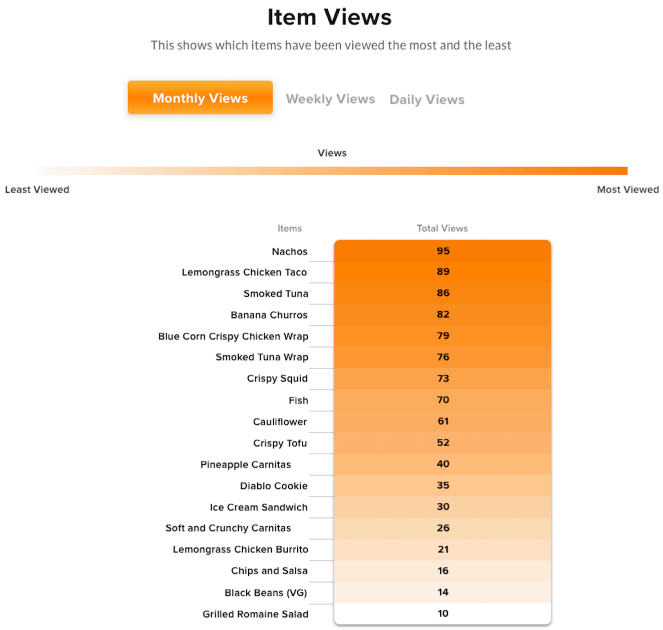
The ambiance of restaurants are quite different from each other. It is useful to consider only ones which have similar ambiances to find similar restaurants. People tend to stay in places that have better ambiances for a longer time. Therefore, the average time spent by customers could be an indication of the ambiance of a restaurant. We use the average time spent by customers as another factor to find similar restaurants as these two factors are highly correlated. The time spent in a restaurant for each customer equals to the difference between the time that users scan the barcode through the mobile app, and the time that they pay for they order.

(VI) Visualization

For this project, we designed four views to help restaurant owners better manage their restaurant. The names of these views are as follows:

- Heatmap view
- Customer-Loyalty view
- Performance view
- Item views
    - (a) Item-sales view
    - (b) Item-inventory view

Each of these views is designed to assist restaurant owners in different tasks. Initially, we designed a mock-up for each of these views to get feedback from some restaurant owners, and then we implemented the ones which we found to be useful.



Item Views

This shows which items have been viewed the most and the least

Monthly Views | Weekly Views | Daily Views

Views

Least Viewed — Most Viewed

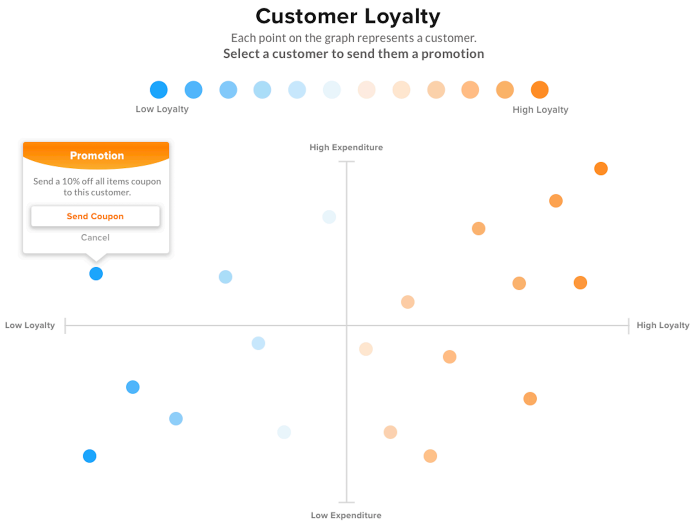
Items	Total Views
Nachos	95
Lemongrass Chicken Taco	89
Smoked Tuna	86
Banana Churros	82
Blue Corn Crispy Chicken Wrap	79
Smoked Tuna Wrap	76
Crispy Squid	73
Fish	70
Cauliflower	61
Crispy Tofu	52
Pineapple Carnitas	40
Diablo Cookie	35
Ice Cream Sandwich	30
Soft and Crunchy Carnitas	26
Lemongrass Chicken Burrito	21
Chips and Salsa	16
Black Beans (VG)	14
Grilled Romaine Salad	10

Figure 1- Heat map for the menu (Mock-up)

(A) Heatmap view

Heatmap view (Fig. 1) is designed to show restaurant owners which parts of their menu have been clicked more than the other parts. The logic behind that is to help restaurant owners design a better menu by positioning their profitable items into the positions which people tend to click more. The initial idea to design this view was to have a heatmap on top of restaurants' menus.

The reason for choosing a 1D heatmap is that users can easily identify the parts that have been clicked more than the others. Also, the one-dimensional representation of the heatmap is aligned to the way that the mobile app is designed.



Customer Loyalty

Each point on the graph represents a customer.

Select a customer to send them a promotion

Promotion

Send a 10% off all items coupon to this customer.

Send Coupon

Cancel

Figure 2- Customer Loyalty View (Mock-up)

There are two attributes in this view: one ordinal and the other one numerical. The ordinal attribute is the rank of that spot on the menu. The value shows how many times that spot has been clicked. The value of each cell is encoded using the channel of luminance, which means the darker cells are clicked more by users.

(B) Customer loyalty view

Customer Loyalty view is designed to assist restaurant owners have a better understanding of their customers in terms of their loyalty and expenditures. Based on that, they can send different promotions to different categories of customers.

What we designed initially for the mock-up (Fig. 2) is a scatter plot in which the X-axis shows the loyalty to that

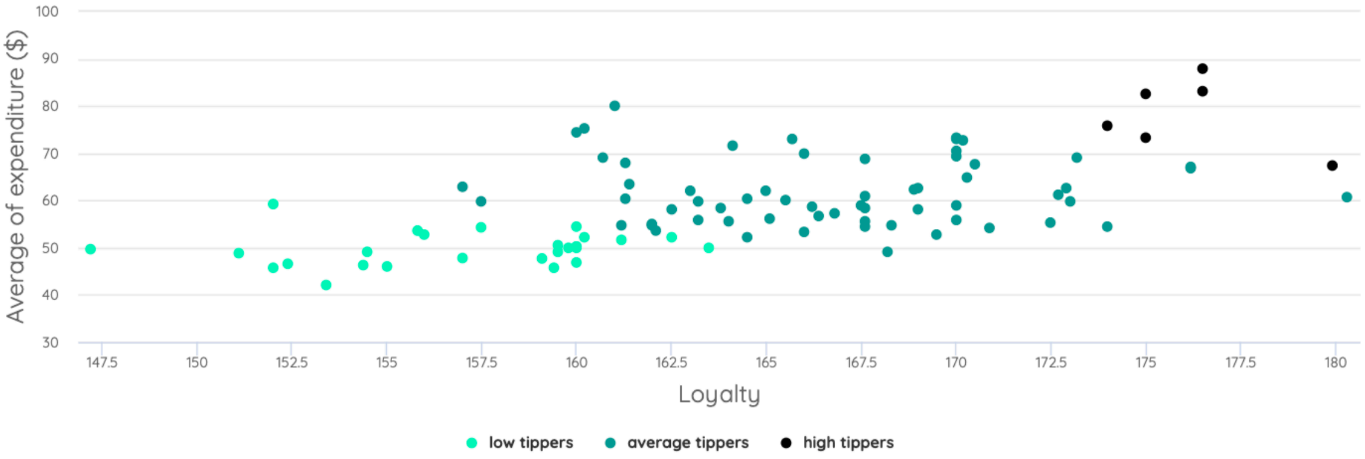


Figure 3- Customer Loyalty View (Implemented)

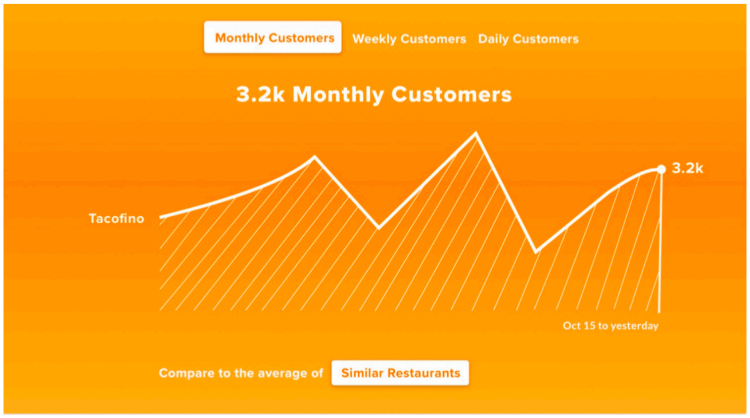


Figure 4- Performance View (Mock-up)

specific restaurant, and the Y-axis shows the average expenditure of the customer across all restaurants. The points in this scatter plot represent customers of that restaurant (Fig. 3). To give an intuition, people who are in the top left corner of this view pay more on average for food, but they are less loyal to that specific restaurant. Therefore, they seem to be good options for sending promotions.

In the mock-up version, we encode the loyalty with the channel of hue, luminance, and spatial position. Using only one of these channels to encode loyalty is enough. Instead, we use the channel of hue and luminance to encode the amount of tip each customer pays and categorize customers into three categories of low tippers, average tippers, and high tippers. For loyalty, we only use the channel of spatial position.

It is beneficial for restaurant owners to find loyal customers as they can offer particular advantages to them. However, is it reasonable to offer special advantages to "all" loyal customers? Customers' expenditure is another dimension which helps restaurant owners classify customers even further, which helps find more profitable customers. We show these two attributes by the X-axis and the Y-axis to make it easy for users to classify customers. The other dimension that helps classifying customers is the amount of tips they give on average. We encode it using the channel of luminance in point marks. We use the spatial position for the measure of loyalty and

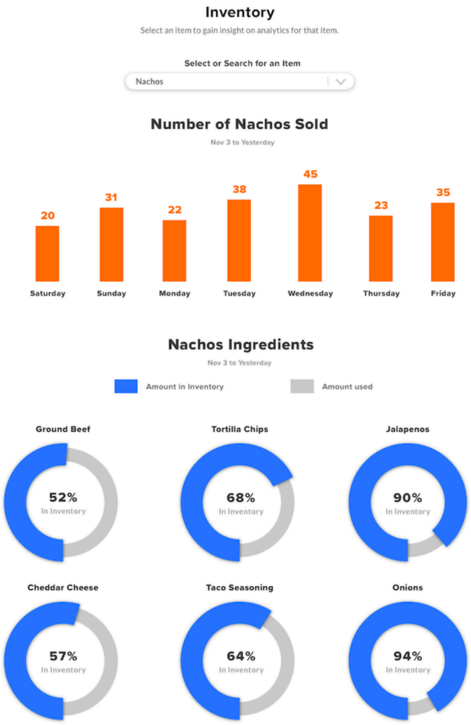


Figure 5- Item view (Mock-up)

expenditure because they have a higher priority. The amount of tips has a lower priority; therefore, we encode it with the channel of luminance.

(C) Performance View

This view is designed to help restaurant owners track their performance in the last two weeks as well as letting them compare their restaurants with the average of similar restaurants. We use the average of similar restaurants due to the confidentiality issues rise from using only similar restaurants

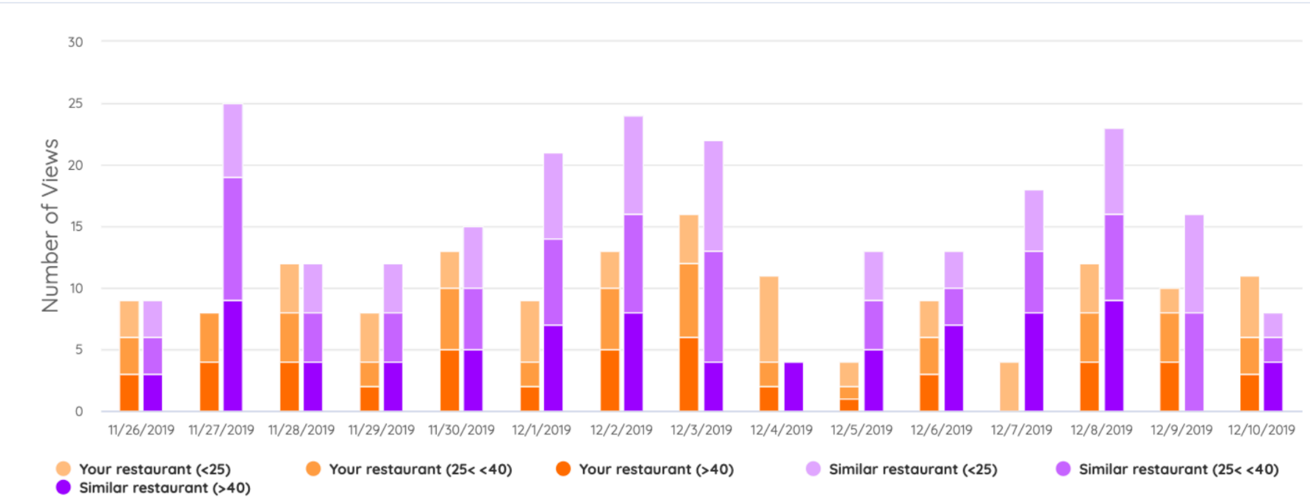


Figure 6- Performance View(Implemented)

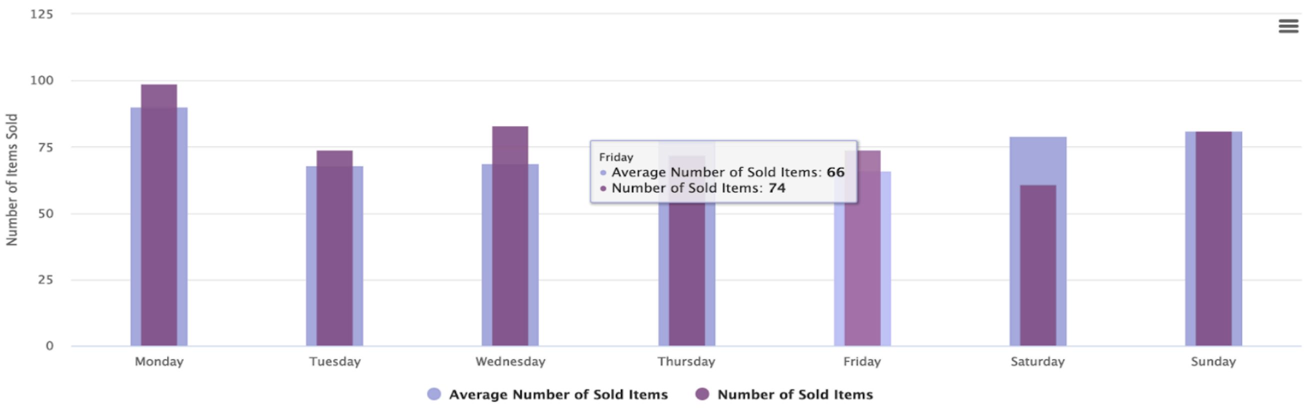


Figure 7- Item sales View (Implemented)

as restaurant owners would not be happy if we share their information. In the mock-up version, we used a line chart, the X-axis of which shows the dates in the last two weeks and the Y-axis shows the number of orders through the mobile application (Fig. 4).

For implementation, we ended up using a stacked bar chart to incorporate customer demographics into the chart as well (Fig. 6). As a result, we are juxtaposing the stacked bar for the restaurant by the stacked bar for similar restaurants. This is done for every single day in the past two weeks. In this view, we use the channel of luminance to show the demographics, and we use the channel of hue to differentiate the restaurant and similar restaurants. By demographics, we mean categorizing customers into three categories of under 25 years old, between 25 and 40 years old, and older than 40 years old. Besides, this view allows restaurant owners to filter out some of the subcategories and compare the rest by clicking on the labels which are positioned below the view. Another view is also designed, which is exactly the same as the previous one except for its Y-axis, which represents the number of views by users through the mobile application, instead.

The reason we choose the stacked bar chart for this view is that it helps to break down the data based on customer demographics. By juxtaposing stacked bars of similar restaurants to the user's restaurant, the comparison with similar restaurants can be made easily. The only problem with stacked bar charts is that each stack starts with a different y value. In some cases, this makes the comparison tricky, and we solved this issue by letting users filter out different categories.

(D) Item views

Item views aim to show all the information about each food item in just one look. There are two views for this section: Item-sales view and Item-inventory view.(Fig. 5) In the very top part of this section, you can filter the food item, then all the data related to that item get updated in both views. By using the filtering, the area of focus will be reduced to just one specific item.

i) Item-sales view

This view shows a bar chart comprising of two types of bars: (1) the number of sold items in the past week. (2) The average number of sold items in total.(Fig. 7) These bars are superimposed over each other to let users compare how well they have been selling their items recently to the average for a more extended period. This comparison can show any anomalies in the popularity of that specific item. The channel of luminance is separating these two bars as the darker bar shows the data for the previous week. This bar is likely to be of more importance to users, therefore, darker color is assigned to it to attract users' attention.

This bar chart offers some interactivity to filter data. Users are able to click on the labels of each bar below the chart to select or deselect them. In this way, users can remove one of the bars and solely view the other one. This helps to reduce the visual clutter and users would find patterns more easily in each graph. Besides, if users need to know the exact number of sold items for each day, they can hover over a bar to see detailed textual information for that specific day.

ii) Item-inventory view

This view shows detailed information about the ingredients of a food item as well as all the ingredients used in the restaurant.(Fig. 8) The view uses a pie chart for each ingredient, and the value of pie charts shows how much of that ingredient is left. Here, we are using pie charts to show consumed and remaining amounts of ingredients based on the maximum capacity of each container which are linearly related. We make use of the channel of hue to attract users' attention to the remaining value. The remaining values are encoded by the hue of red and blue and the consumed values are encoded by the hue of pale gray. We are using the mentioned blue and red hues to encode the emptiness of the containers. Red is used for ingredients that are running low and blue for other ingredients. For ingredients which are not running low, other than using

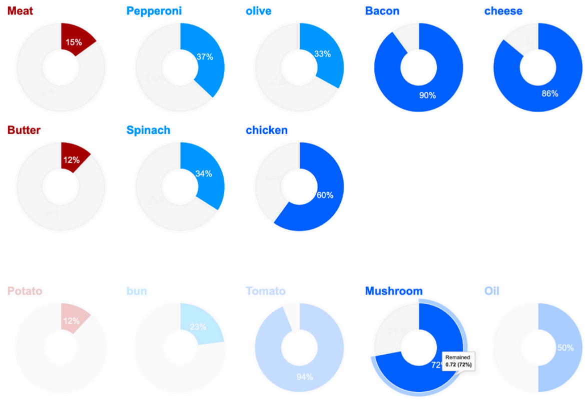


Figure 8- Item inventory View (Implemented)

blue as their hue, the channel of luminance separates them based on full their containers are. These values get updated automatically as customers place their orders. As we are knowledgeable about the ingredients of each food item, we will be able to update these amounts based on the orders.

In this view, users are able to scroll down to inspect all the ingredients that are used in the restaurant. The channel of spatial position in the 2D space and the channel of luminance help users find related ingredients to the selected item. As we select an item, all the ingredients related to it become more brighter while other ingredients fade out. Also, the related ingredients would be positioned on the top of the view so that users can spot them faster and easier.

Although pie charts consume a lot of space for showing a few information for each ingredient, it is more intuitive for users as it is really understandable for most people. We could have used the channel of spatial position more efficiently for finding empty containers by sorting the containers by their emptiness. However, this method would have drawn users' attention to empty containers too much, while they should consider full containers and unused containers as well.

(VII) Alternative Visualizations

To provide an overview for inventory management, we also considered some other views which consume less space. All of them were problematic in some ways and therefore, we decided not to implement them, and here we just discuss their deficiencies.

The first view is a hierarchical tree that has the ingredients as its leaves and the food items as its = internal nodes. It shows the family of each food item using line marks, and if you hover over a node (a node can be an ingredient, a food item, or a food category), all the family is shown as well as the information for relevant ingredients. Leaves can be color-coded by their value and also, internal nodes can be color-coded by the average of their ingredients' values. With this view, you can monitor the inventory all in one place.

This visualization is problematic due to the following reasons. First, the hierarchy is not as straightforward as it is in Figure 9. The food family tree would not be a binary tree.

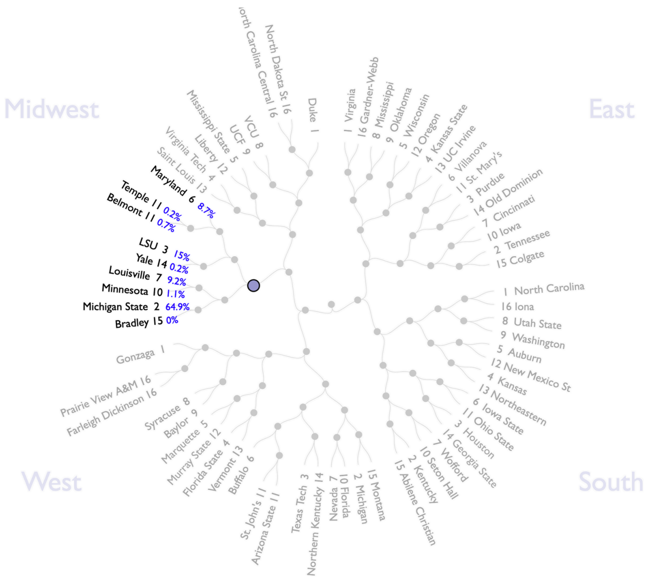


Figure 9- The binary tree architecture for basketball men competition. This is an example of a hierarchical tree in a different context.

Some ingredients are common among most of the food items, and connecting them by using line marks could cause visual clutter. Although aggregating lines with thicker lines is not a bad idea, it still can not solve the complexity of the food hierarchy. Second, it is hard to gather the data in a way to be presentable by this structure. The last issue with this view is the scale problem. The number of ingredients and food items can not pass a certain number which is not ideal.

The second view is a 2D heatmap (like Fig. 10) that presents food items on the Y-axis and ingredients on the X-axis. Each cell shows the remaining amount of each ingredient for each food item using the channel of luminance. If you hover over a food item, the related row would become brighter.

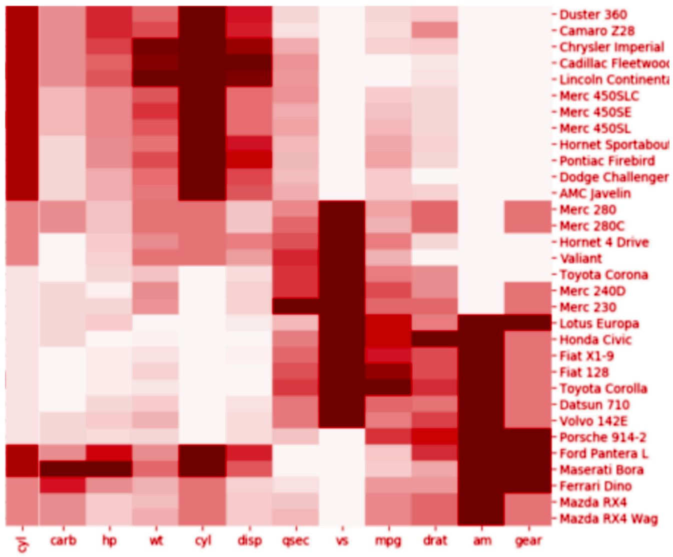


Figure 10- 2D heat map for two categorical attributes. Using the channel of luminance to encode values for each cell.

All cells in one column would be similar as they are encoding the same ingredient. They are consuming a lot of space to show one tuple of data. Although a 1D heat map could solve this issue, it could not show the related ingredients to each food item easily.

Each food item does not have all the ingredients, so lots of cells should be inactive. This is not the most efficient way to use the pixels on the screen.

(VIII) IMPLEMENTATION

(1) Data Schema Manipulation

When the implementation started, the database was prepared for only the first milestone of the project, which was the digitized menu. We did not have the tables for the ordering feature yet. The tables related to the ordering feature is critical for some of our tasks such as loyalty identification and item sales overview. Therefore, we had to add these tables to the database to be able to start coding.

(2) Data Synthesis

We do not have a real data, as real users have not started using our system yet due to the fact that the mobile application is not up and running. Therefore, after having all the necessary tables in our dataset, we started to fill the database with some dummy data. All the data synthesis processes follow the main data schema to make the integration of the visualization tool into the web application possible in the future. We synthesized data for items, ingredients, and restaurants. Moreover, we synthesized some customer orders data to address tasks about loyalty and sales analysis.

(3) Normalization

We use normalization in our system two times. The first one is the normalization of ingredients' data, and the second one is the normalization of the loyalty measures.

Since there are various metrics for measuring the ingredients, we should normalize the ingredients' remaining values. Our basic approach for the normalization is to map the remaining value of each ingredient to the range [0,1]. By doing this, we can show the percentage for the remaining ingredients, which is reasonably intuitive for restaurant owners.

Moreover, we do the same kind of normalization for the loyalty measure, which is the last step in the loyalty algorithm. The algorithm itself does this task and output a number in the range [0,1].

(4) Creating Mock-up Views

After gathering requirements and abstracting all the tasks, we started to generate mock-up views for each task. A mock-up view is just a design of how the view would look like to address a specific task.

(5) Focus Groups

To refine the mock-up views we conducted a focus group to gather feedback from real users. We shared those mock-up views with restaurant owners, and they commented on each of them. We will discuss the results of this step in the evaluation section.

(6) Coding

We started the coding process after we had all the refinements done on our views. All the views are implemented in the web environment using HTML, Javascript, and CSS. Performance and customer-loyalty views are implemented with the React JS platform, while item view is implemented with raw web code.

(6-A) Tools:

We choose our tools based on the requirements for each visualization component used in the mock-up views. Below is the list of requirements we needed for all of the views:

(a) For the sales bar chart in the item view, we need to juxtapose the bars. Plus, we need to superimpose the average and the current data over each other.

(b) For the sale bar chart in the item view, we need to do filtering on data. To select/deselect average and current data.

(c) For the loyalty chart, we need to select data points.

(d) For the stacked bar chart in the performance view, we need to be able to juxtapose different bars and color code different stacks.

(e) For the inventory management view, we need to arrange pie charts as the user selects different items. We need to configure pie charts dynamically, such as changing hue and luminance level.

(f) For all the charts, we need to customize colors to address our tasks.

Highcharts can address most of the mentioned requirements, therefore, we choose it as our primary tool. However, it can not address item (e), as we can not use its pie charts dynamically. Thus, we use Google Charts to approach this requirement.

(IX) EVALUATION

Before implementing the views, we validated our ideas by conducting a focus group with a number of restaurant owners, some people who are experienced in the restaurant industry, and the appetize team. In that focus group, we shared the mock-up views with those people and got their feedback for each of the views.

For the Heatmap view, we found out that this view is not extremely useful for restaurant owners as they already knew that people tend to choose those items which are positioned at the beginning of a menu. Therefore, we decided to eliminate that view from our list and not to implement it anymore.

The focus group had two main takeaways for the Customer-Loyalty view. First, we do not need to use the diverging mode for both the Y-axis and X-axis as all the numbers fall into the positive range. Besides, we realized that we could use the channel of hue and luminance for other goals other than encoding loyalty as we already encoded that by leveraging the channel of spatial position. Therefore, in the implemented version, we only show the positive ranges for both axes and we use the channel of luminance and hue for another purpose, which is to categorize customers into three categories of low tippers, average tippers, and high tippers.

Performance view did not have as many comments as the previous views except they believe it would be more beneficial to have the customer demographics incorporated into the view. This will help them with marketing purposes as well as giving them a better understanding of their customer personas. As a result, for implementation, we used a stacked bar chart instead of a line chart.

(X) Discussion

(A) Limitations

Even though our visualization system gives insightful information to restaurant owners, it has some limitations too. The most important one is that all the visualizations are based on dummy data. The mobile application is not up and running, therefore, customers did not have the chance to use the mobile app. Therefore, we had to synthesize data to be able to design these views. In the performance view, we only allow restaurant owners to compare the performance of their restaurants with the average of similar restaurants within the last two weeks, which is not ideal as they might want to analyze for other times or more than two weeks.

Inventory management view does not show an overview of all food items, which makes restaurant owners choose each item from the drop-down list one by one to see the relevant ingredients. This may take time for restaurant owners, especially those who have a restaurant with more variety of food items. Besides, we are consuming a lot of pixels to show the ingredients and the emptiness of ingredient containers, which could have been done more efficiently.

In addition, the algorithms are not still implemented and there might be some challenges showing up after the implementation of algorithms.

(B) Future work

For the future work, we will implement the algorithms which are mentioned in the algorithm section. Moreover, we will try to focus on our limitations, solve them or make them less problematic. Finally, we will design more visualizations to cover those abstracted tasks that we have not gotten the chance to cover in this project such as visualizing the profitability of food items (Task #1).

References

1. Improving Service Processes Based on Visualization of Human-Behavior and POS Data: A Case Study in a Japanese Restaurant; Tomohiro Fukuhara, Ryuhei Tenmoku, Takashi Okuma, Ryoko Ueoka, Masanori Takehara, Takeshi Kurata, 2014.
2. Anil Bilgihan, Soobin Seo & Jihee Choi (2018) Identifying restaurant satisfiers and dissatisfiers: Suggestions from online reviews, Journal of Hospitality Marketing & Management, 27:5, 601-625.
3. Jurui Zhang. (2019) Listening to the Consumer: Exploring Review Topics on Airbnb and Their Impact on Listing Performance. Journal of Marketing Theory and Practice 27:4, pages 371-389.
4. Ricardo Godinho Bilro, Sandra Maria Correia Loureiro, João Guerreiro. (2019) Exploring online customer engagement with hospitality products and its relationship with involvement, emotional states, experience and brand advocacy. Journal of Hospitality Marketing & Management 28:2, pages 147-171.
5. Tao Zhang, Junyu Chen, Baoliang Hu. (2019) Authenticity, Quality, and Loyalty: Local Food and Sustainable Tourism Experience. Sustainability 11:12, pages 3437.
6. C.M. Sashi, Gina Bryníldsen, Anil Bilgihan. (2019) Social media, customer engagement and advocacy. International Journal of Contemporary Hospitality Management 31:3, pages 1247-1272.
7. M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.