

menuVis: a Recipe Creation Visualization Tool for Chefs

Silver Burla

silverbu@cs.ubc.ca

CPSC 547 Information Visualization
The University of British Columbia

Margot Chen

c3xiaoy@gmail.com

CPSC 547 Information Visualization
The University of British Columbia

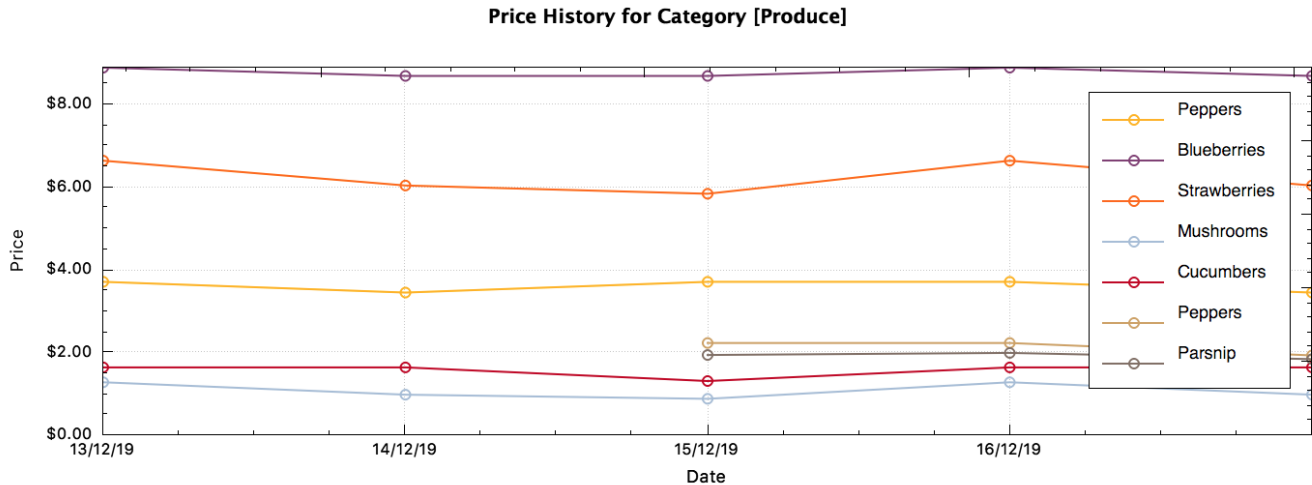


Figure 1: Ingredient price history over the Produce category.

ABSTRACT

We present menuVis, a recipe creation tool that helps visualize and analyze food cost for chefs. We collected two user requirements: (1) creating and adjusting recipes, and (2) monitoring the volatility of ingredient cost. Based on these two requirements, we provide visualization solutions and embed them into chefs work flow.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**; **Information visualization**; • **Applied computing** → *Enterprise data management*.

KEYWORDS

infovis, interactive tool, restaurant, business, costing

1 INTRODUCTION

Running a restaurant is not easy. It requires delicate balance between making customers happy and keeping the menu profitable. In the United States, restaurants run by independent owners (two or fewer) had a three-year failure rate of 61.4%[4]. To keep their businesses vigorous, chefs create new recipes on a regular basis and spend considerable time on recipe costing.

In this paper, we present menuVis, a small but effective recipe managing tool that helps visualize the cost benefit analysis of recipes on a wide spectrum of granularities, using modern data visualization techniques. In designing specifically for the tasks of creating, editing, and evaluating recipes, menuVis will help chefs better leverage the data available to them, so that they may design

more effective menus, and spend more time on creating delicious recipes instead of book keeping.

In collaboration with chefs from three food service businesses in Vancouver, we collected data and requirements from potential users. In the course of designing menuVis, we had qualitative discussions with chefs on a variety of visualization idioms. To refine our design, we conducted a qualitative usability test with chefs.

We begin with a brief review of related work in section 2. In section 3, we describe the data abstraction and user requirements. We discuss our visualization solution in section 4 and implementation details in section 5. We walk through menuVis using a scenario of use, and evaluate our design through a usability test in Section 6. We conclude and discuss the future work in Section 7 and 8.

2 RELATED WORK

Optimum Control¹ is a restaurant inventory management software. It is used by one of three participating businesses and is allegedly the cheapest among alternative software, while also operating under an expensive subscription model. It has recipe creating, costing, ingredient purchasing, and other functions in one platform. It generates more than 70 reports on inventory, ingredient purchasing, sales, recipe, and other topics, among which the most frequently used reports by the participating chef are Recipe Detail Report, Menu Engineering Worksheet (see Appendix 1), and Menu Product Detail Problems Report.

The Recipe Detail Report contains names, amount, and cost of each ingredient used in a recipe. The Menu Engineering Worksheet reports a menu product's popularity and profitability over a week,

¹<https://tracrite.net/>

a month, or a customer-defined period. The Menu Product Detail Problems Report compares the desired cost of products with the actual cost and indicate products whose actual cost is problematic.

All reports are tables. There is no visualization of the results. Users need to read them closely to locate potential issues. For the Menu Product Detail Problems Report, users have to check out each ingredient of a product, switching between multiple windows to find out the cause of cost change.

Our visualization was inspired by Optimum Control in terms of understanding user requirements. Users want to keep a record of ingredients, cost, and profit margin of recipes. They also want to monitor the volatility of cost and keep their products profitable. Instead of making users read long tabular reports and shuffle between windows, we aim to use visual encoding and interaction to help users obtain insights about their recipe. We also aim to embed visualization in chefs' work flow, so they can get these insights in a natural and effortless way.

3 DATA & TASKS

3.1 Domain Background

We gathered data from a food prep and delivery service, a local bar-and-grill, and a local diner. The data from each business are formatted differently and at varying granularities, reflecting the difference in both management styles and business strategies between them. The basic data they have in common are monthly sales, recipes, and ingredient pricing.

Recipes are written in a tabular form, containing information of recipe names, recipe categories, volume of each ingredient, the cost of ingredients, and the price of the recipe. A restaurant can hold 10-100 recipes at a time.

Ingredient pricing is tabular data gathered from food suppliers such as Sysco and GFS. It contains name, category (such as meats, dairy, and produce), supplier, price (by amount), and pricing date of each ingredient. The food suppliers provide order guides with pricing information. Varying from supplier to supplier, there may be 5-10 categories, and 100-400 specific ingredients in the order guide. The ingredients that are actually ordered by a restaurant are a subset of ingredients available on the order guides.

When creating a new recipe, chefs first come up with a theme of the recipe and set the price and desired percentage of cost. Surrounding the theme, chefs then pick ingredients and decide the amount of each ingredient, keeping in mind that the cost of ingredients should be no more than the desired cost. The cost of ingredients, especially produce, is changing all the time. Therefore, after creating and costing a recipe, chefs need to check the actual cost and make adjustments to the recipe frequently.

3.2 Data

The attributes of recipe data and ingredient pricing data are described in Table 1 and Table 2.

3.3 Tasks

We conducted semi-structured interviews and contextual inquiries with the owner and chefs of the three participating businesses to gather their requirements. Based on the interviews and inquiries, we gathered two user requirements:

Requirement 1. Creating and adjusting recipes based on ingredients cost. Keeping food cost rate low is vital to the survival of a restaurant. Two out of three participating chefs said food cost was the indicator they cared most when creating a new dish.

Requirement 2. Monitoring the volatility of ingredient cost and calculate real-time cost of recipes. All the participants noticed the fluctuation of ingredients cost, and had to make adjustments to their recipes based on the cost change.

In abstracting the tasks, our goals are to present, compare, and summarize our data. For individual recipes, we want to present the ingredients and cost of each ingredient, and also compare the cost with the desired cost set by user. For ingredients cost which has hundreds of items, we want to present the price of individual items over time, and also summarize the cost volatility of all the items.

4 VISUALIZATION SOLUTION

In this section, we describe our visualization solution summarized in Table 3, and analyze it in the what-why-how framework by Munzner[3].

menuVis has two tabs, the Ingredients Tab and the Recipes Tab. The Recipes Tab allows users to create recipes, and inspect the composition and cost of recipes. In Ingredients Tab, we present the ingredients cost over time.

4.1 Ingredients Tab

We use a line chart (Figure 2) to show the price history of ingredients. When users select one ingredient, the price history of ingredients in the same category, such as proteins, produce, or dairy, will show up, with the selected ingredient highlighted. The bar chart (Figure 3) shows the percentage of cost change of the selected ingredient.

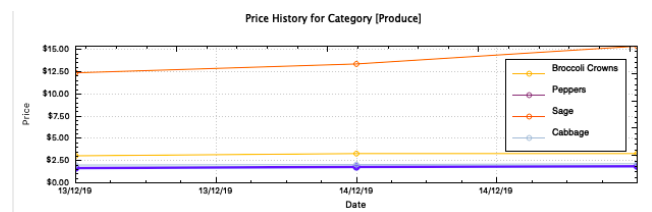


Figure 2: Price History for Category.

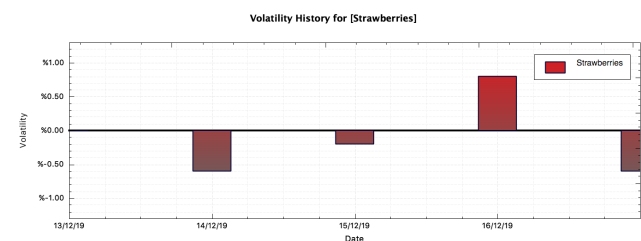


Figure 3: Volatility History for Selected Ingredient.

There can be hundreds of ingredients in use at a time in a restaurant. But when an ingredient becomes too expensive, chefs look for its substitute in the same category. Thus, we bin ingredients

Attribute Name	Attribute Type	Description	Level/Range
Name	Categorical	The name of a recipe.	10-100
Category	Categorical	The category of recipes ² .	3-8
Ingredient	Categorical	Ingredients used in a recipe.	<10
Amount	Ordered	The amount of each ingredient used in a recipe.	0.1-1,000
Unit	Categorical	The unit of measurement of ingredients ³ .	<10
Cost	Ordered	The cost of ingredients given the amount used.	<10
Price	Ordered	Price of the recipe.	1-50

Table 1: Attributes of Recipe Data

Attribute Name	Attribute Type	Description	Level/Range
Supplier	Categorical	The supplier of ingredients.	1-10
Category	Categorical	The category of ingredients ⁴ .	5-10
Name	Categorical	The name of an ingredient.	100-400
Description	Categorical	The description of an ingredient.	100-400
Brand	Categorical	The brand of an ingredient.	50-300
Cost	Ordered	The price of an ingredient per unit.	2-500
Item Number	Categorical	The unique identifier of an ingredient.	100-400
Pricing Date	Ordered	The date when the price is generated.	A year

Table 2: Attributes of Ingredient Pricing Data

Requirement 1	
What: Data	Recipes (multi-dimensional table).
Why: Tasks	Present and compare.
How: Encode	Stacked bar chart to present the cost of ingredients of a recipe, and compare the total cost with the desired cost.
Requirement 2	
What: Data	Cost of ingredients over time (time-series table).
What: Derived	Volatility of individual ingredients.
What: Derived	Aggregated volatility of a recipe.
Why: Tasks	Present and summarize.
How: Encode	Bar chart to present cost volatility of individual ingredients; Line chart to present cost of multiple ingredients over time; Bar chart to show aggregated volatility of recipes cost.
How: Reduce	Aggregate volatility of ingredients cost into volatility of recipe cost; Select ingredients of interests.

Table 3: What-Why-How Analysis of menuVis

by category. In a category, there can be dozens of ingredients. To mitigate the potential occlusion with dozens of lines, we highlight the selected ingredient.

We also tested heatmaps with two chefs. The two axes of the heatmap are ingredients and pricing dates. The percentage of change of ingredients cost is encoded in a blue-red diverging color scheme. Two chefs said they could understand the heatmap. But when comparing the heatmap with line chart and bar chart, chefs showed preference to the latter two. In order to adjust their recipes promptly, chefs need to know the exact change of ingredients cost, rather than the big-picture trend or pattern of the change. Color encoding is not suitable for showing the precise numbers.

4.2 Recipes Tab

The cost of a recipe is shown in a stacked bar chart (Figure 8). The ingredients of a recipe are color-encoded, while the cost of each ingredient is length-encoded. The length of the entire bar represents the total cost of a recipe. Users can change the ingredients and the amount of each ingredient. The graph will update according to users' choices.

Once a recipe is created, it can be viewed in the Recipes Tab. Again, we use a line chart to show the price and cost history of a

²such as salad, breakfast, and entree.

³such as g, kg, oz, ml, l.

⁴such as meats, dairy, and produce.

recipe, and a bar chart to show the volatility of cost of the selected recipe. The change of ingredients cost is aggregated into recipes and shown in both the line chart and the bar chart. The equation used to calculate recipe volatility measures the total expected percentage of the recipe cost a chef can expect to increase or decrease, within the given time frame since their last pricing guide, or costing, update.

$$Recipe_{volatility} = \frac{\sum(ingredient_{volatility} * ingredient_{amount})}{\sum(ingredient_{\$/uom} * ingredient_{amount})}$$

We use a nested pie chart shown in Figure 4 to present the components of a recipe. The amount of ingredients in each category, such as proteins and produce, are encoded by angle in the pie chart. The breakdown of ingredients in each category is shown in the concentric circle around the pie chart, distinguished by saturation. The position of the concentric circle and the pies show the part-whole relationship of ingredients and category. When hovering on an ingredient on the concentric circle, the information of this ingredient will be shown beside the chart.

The composition of a recipe is further discussed in Section 7.

The number of discriminable steps for hue is around six or seven bins[3, p. 224]. There can be up to 10 ingredients in a very complicated recipe. We use Kenneth Kelly’s[1] color list of maximum contrast in the stacked bar chart to make each ingredient distinguishable.

5 IMPLEMENTATION

5.1 Development Environment

The menuVis desktop application was coded in C++ using QT as the primary toolkit. QT is a cross-platform application development framework written in C++, and an open source widget toolkit for creating graphical user interfaces. R was our secondary toolkit used specifically to prototype potential visual encoding idioms, and test them with participating chefs. For the visualizations themselves, QT Charts and QCustomPlot were used as they integrated well into the QT framework, and offered a good level of control over appearance and interactions. For reading spreadsheet files, Qxlsx was used to remove the excel proprietary wrapper around .xlsx files, and enable easier parsing of their internal JSON format.

5.2 Development Stages

The development process can be broken into three main stages; user interface coding, database coding, and coding visualizations.

5.2.1 User-Interface. Coded first to provide chefs with an idea of what the tool may look like, and aid us in probing them for further requirements. The QT framework did the majority of the work during this stage, as it supplied its own GUI drag and drop workspace for building graphical interfaces with. However, the QT support provided here was for the laying out of boxes and labels, the actual linking, messaging, and event handling between interface elements were coded by hand. Additionally, custom popup windows were coded outside the provided drag and drop GUI, and dynamically created interface elements were required in the recipe creation window. The user interface was continually refined during

the full length of development process, with a strong focus towards guarding the database against malformed user-created data.

5.2.2 Database. The database was coded without the aid of any toolkits. Given that menuVis must handle continuously growing data, with each new pricing guide uploaded from suppliers, user created ingredient, and recipe creation, there are many potential sources of malformed data; either an error on the part of the suppliers pricing guide, or the user creating bad data by mistake. Since the purpose of the application is to accumulate data and evolve with use, data integrity and constancy were extremely important factors in deciding the structure of the database, and how it would fit into the larger scope of the application. The Singleton design pattern was used to create and maintain a single data delegate instance, shown in orange in Figure 5, which is responsible for providing controlled access to both ingredient and recipe data, shown in blue. The Model-View-Controller design was adapted to the Model-View structure of the menuVis codebase using models, shown in green, and viewers shown in red. By using signals and slots, the QT versions of event emitters and listeners, viewers are decoupled from the underlying data. Viewers must go through their respective models, which submit specific read and write requests to the data delegate, to interact with, combine, derive, and display any data.

5.2.3 Visualizations. QT Charts and QCustomPlot were used to create the visualizations of the data. In total, there are six visualizations integrated into menuVis. While both QT Charts and QCustomPlot provide support for creating graphs, the visualization needs of menuVis required extensive customization across each graph. For example, the visualizations needed to update in real time with data as it grew, selections of ingredients in the ingredient table view needed to reflect onto the price history visualization, hover events needed to be transmitted for ingredient data to be retrieved and displayed in the recipe viewer, and volatility and price per unit metrics needed to be calculated, saved, and all windows and visualizations notified.

6 RESULTS

menuVis is designed for creating recipes and monitoring cost change. In this section, we present menuVis through a scenario of use and provide an evaluation of menuVis.

6.1 Scenario of Use

When create a new recipe, chefs first need to get the latest pricing information of potential ingredients from the order guides provided by suppliers. Clicking the update button in the Ingredients Tab (Figure 7) will open the Ingredient Staging Area shown in Figure 6. Chefs can upload and edit the latest order guide here. Since different suppliers categorize ingredients in various ways, menuVis requires that users assign a category to each ingredient manually for clarity and consistency.

Once ingredients are saved in the system, if users upload a new order guide, menuVis will automatically detect the existing ingredients and both disable and gray them out, so that users need not to select them and assign categories to them again.

In the Ingredients Tab (Figure 7), when selecting one ingredient, the price history for ingredients in the same category will be shown

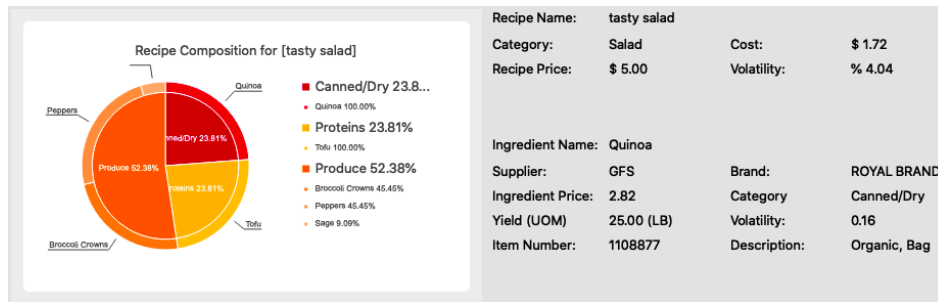


Figure 4: Recipe Composition.

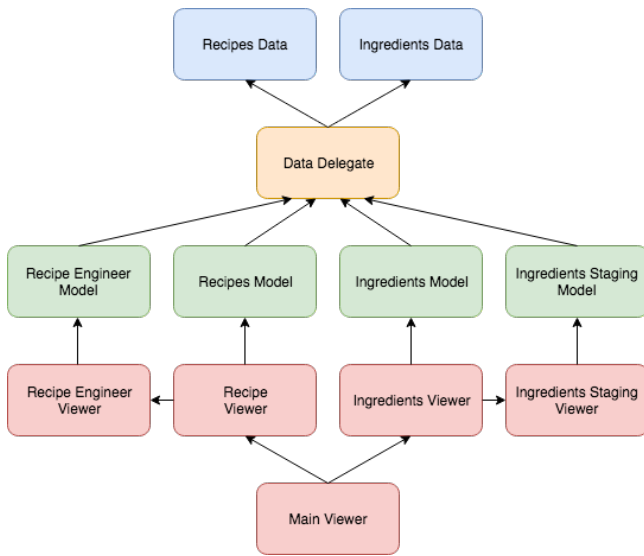


Figure 5: menuVis database diagram and data flow for main classes, showing the "has-a" relationship implemented. Helper classes and dynamic viewers are not shown.

in the line chart, and the bar chart will show the volatility of the selected ingredient.

Now chefs can create recipes with existing ingredients in the Recipes Tab. Clicking Add opens the Create New Recipe window (Figure 8). When selecting one ingredient, the text in the bottom-left corner will show how much of this ingredient is used in how many recipes. This will give chefs a sense of the usage of the ingredient. Chefs can add ingredients in the left panel, and adjust the amount, choose the unit they prefer, and set the price of the recipe on the right panel. The composition and total cost of the recipe will be shown in the stacked bar chart.

Once a recipe is saved, it can be viewed in the Recipes Tab (Figure 9). When the ingredients cost is updated, menuVis will detect the change of cost and aggregate it into recipe volatility. When a recipe is selected, the line chart will show the price and cost history of the recipe, and the bar chart volatility history. In the sample recipe in Figure 9, chefs can notice that the volatility of "tasty salad" has increased 4%. When hovering on the ingredients in the nested pie chart, the breakdown volatility of each ingredient

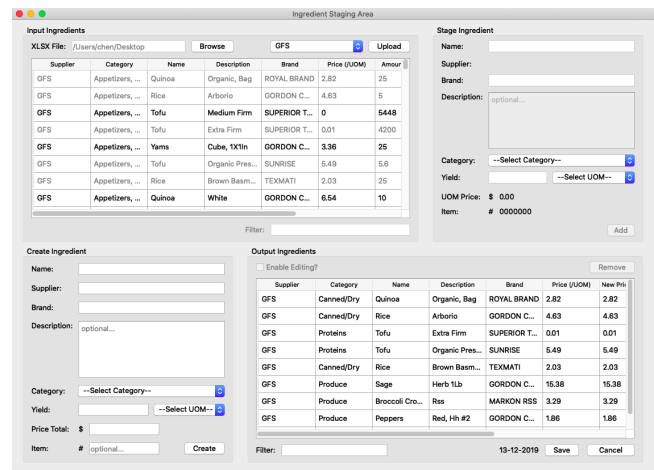


Figure 6: Ingredient Staging Area.

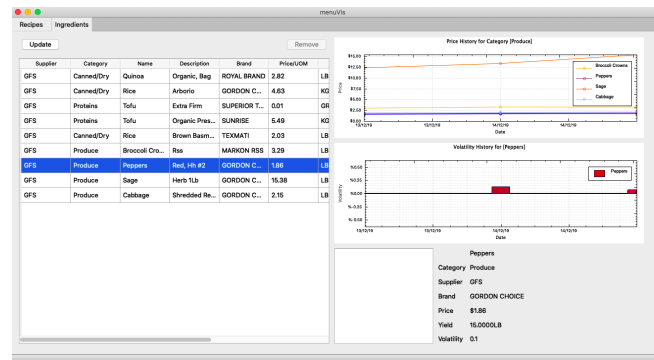


Figure 7: Ingredients Tab.

used in the recipe will be shown. Chefs may notice the volatility of basil is high and go back to the Ingredients Tab to find a substitute for basil.

6.2 Evaluation

Due to time constraints, we conducted a qualitative usability test with two chefs and two sous chefs. Participants were asked to create a recipe on menuVis. We gathered feedback through a concurrent

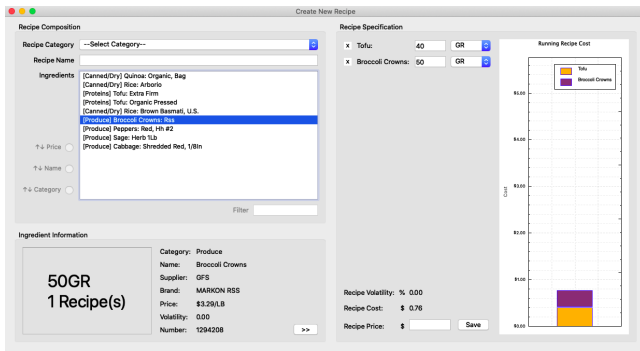


Figure 8: Create New Recipe window.



Figure 9: Recipe Tab.

think-aloud protocol. After the test, participants were asked to fill out a system usability scale (SUS) questionnaire developed by John Brooke[2, p. 447].

During the test, all participants said they found menuVis "very cool", "very convenient". One participant who also uses Optimum Control said he liked that on the menuVis Recipe Tab, all the detailed ingredients information was shown in one place.

One participant was curious about how volatility was calculated.

Two participants pointed out that food was often prepared in batch and distributed to individual servings. They wanted to create recipes and inspect the cost both for batch and for individual servings.

Two participants did not notice that they could hover on the nested pie chart to get ingredients information. But after prompt, they reported that they liked the breakdown of a recipe shown in the chart.

One participant reported he usually set the recipe price and a desired percentage of cost before creating a recipe. For the stacked bar chart, he hoped it could show the difference between the current cost and the desired cost.

We calculated the SUS score according to the formula developed by John Brooke. One participant was in a hurry and chose "Strongly Agree" to every question. The score of other three participants are 80, 77.5, and 95. The participant giving the lowest score expressed a lack of confidence in using this system.

7 DISCUSSION AND FUTURE WORK

The usability test suggests how we can improve the current design. During user research and prototyping, we also collected requirements and ideas that we would work on in the future.

In a recipe, some ingredients can be "prep" ingredients, such as salad dressing designed by the chefs, which themselves also have recipes. Dishes, prep ingredients, and raw ingredients are on different levels of a hierarchical structure. A dish is composed of various prep and raw ingredients. A prep ingredient is made from raw ingredients. When taking preps into account, recipes are in a tree structure, and can be shown in the containment channel.

We tested a prototype of a treemap with three chefs. Two chefs out of three expressed their preference with the treemap. One chef suggested we put a stacked bar chart and a treemap side by side. The stacked bar chart would show the high-level structure of a recipe, and the treemap fine-grained details. Chefs can sense the cost of a recipe through the bar chart, and understand the composition of a recipe through the treemap.

In user research, we noticed that chefs wanted to understand the usage of each ingredient. Chefs re-use the same ingredients in multiple recipes for the sake of easier inventory management. But they also need to keep a variety of ingredients to satisfy capricious customers. Knowing in which dishes a certain ingredient is used helps chefs to plan their menu.

For now the ingredients usage is shown in a text box in the bottom-left corner of the Create New Recipe window (Figure 8). In the future, we plan to use a table of isotypes mapping to the menu. Dishes using the selected ingredient will be marked with corresponding isotypes.

8 CONCLUSIONS

We designed and implemented menuVis, a recipe creation tool that helps chefs create recipes and monitor the cost. In menuVis, visualization idioms are embedded into chefs' work flow. Chefs can inspect ingredients cost in the Ingredients Tab. The Recipes Tab presents the composition of recipes, and the aggregated cost change of recipes, helping chefs to locate the potential issues and make better choices of ingredients.

ACKNOWLEDGMENTS


To the participating chefs and businesses, for trusting us with their data and providing valuable feedback.

REFERENCES

- [1] Paul Green-Armytage. 2010. A Colour Alphabet and the Limits of Colour Coding. *Color: Design Creativity* 5 (08 2010), 1–23.
- [2] Rex Hartson and Pardha Pyla. 2012. *The UX Book: Process and Guidelines for Ensuring a Quality User Experience* (1st ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [3] T. Munzner. 2015. *Visualization Analysis and Design*. CRC Press. <https://books.google.de/books?id=NfKYCwAAQBAJ>
- [4] H. G. Parsa, Jean-Pierre I. van der Rest, Scott R. Smith, Rahul A. Parsa, and Milos Bujisic. 2015. Why Restaurants Fail? Part IV: The Relationship between Restaurant Failures and Demographic Factors. *Cornell Hospitality Quarterly* 56, 1 (2015), 80–90. <https://doi.org/10.1177/1938965514551959> arXiv:<https://doi.org/10.1177/1938965514551959>

A FORMS

A.1 Optimum Control Sample Report



Joess Fine Dining (JOES)

Menu Engineering Worksheet

2/4/20
13:52

From: Tuesday, October 1, 2013
To: Thursday, October 31, 2013
Gross Sales for Period: \$70,110.20

Note: This report calculates a menu product's Popularity (Menu Popularity Factor) and Profitability (Average Item Profit) within the group.

Item Description	Sold	Popularity %	Selling Price	Current Cost	Margin %	Totals			Analysis		Menu Item Class	
						Sales	Cost	Profit	Profitability	Popularity		
Food												
Add-on												
Wi/Soup Cup	323.00	100.000%	\$1.50	\$0.40	\$1.021	31.95%	\$484.50	\$154.78	\$329.72	LOW	HIGH	Workhorse
Average Item Profit: \$1.021 Menu Popularity Factor: 80.00%												
Appetizer												
SPRACH SAL	322.00	100.000%	\$9.50	\$3.10	\$6.398	32.66%	\$3,059.00	\$998.97	\$2,060.03	LOW	HIGH	Workhorse
Average Item Profit: \$6.398 Menu Popularity Factor: 80.00%												
Beverage												
Coffee-Size Cup	640.00	8.954%	\$1.00	\$0.16	\$0.845	15.51%	\$640.00	\$100.50	\$547.50	LOW	LOW	Dog
MILLER LITE	449.00	6.264%	\$3.75	\$0.39	\$3.360	10.49%	\$1,683.75	\$175.11	\$1,508.64	HIGH	LOW	Challenge
Pop-10 oz Glass	5294.00	71.900%	\$1.25	\$0.16	\$1.086	13.12%	\$6,595.00	\$603.40	\$5,991.60	LOW	HIGH	Workhorse
Red Wine-6oz Glass	203.00	3.910%	\$4.25	\$2.87	\$2.182	48.95%	\$1,202.75	\$681.13	\$521.62	HIGH	LOW	Challenge
White Russian	342.00	4.726%	\$4.25	\$0.97	\$3.282	22.77%	\$1,453.50	\$330.92	\$1,122.58	HIGH	LOW	Challenge
White Wine-6oz Glass	311.00	4.297%	\$4.25	\$1.37	\$2.884	32.13%	\$1,321.75	\$424.70	\$897.05	HIGH	LOW	Challenge
Average Item Profit: \$1.429 Menu Popularity Factor: 13.33%												
Combination Dish												
Catering dinner #1	340.00	22.970%	\$18.00	\$5.61	\$12.388	31.18%	\$6,264.00	\$1,952.84	\$4,311.16	HIGH	LOW	Challenge
Cheeseburger Combo	593.00	39.142%	\$4.99	\$1.62	\$3.369	36.49%	\$2,959.07	\$1,077.94	\$1,881.13	LOW	HIGH	Workhorse
Hamburger Combo	574.00	37.860%	\$3.99	\$1.55	\$2.438	38.06%	\$2,290.26	\$689.13	\$1,601.14	LOW	HIGH	Workhorse
Average Item Profit: \$5.812 Menu Popularity Factor: 26.67%												
Dessert												
Peano Nuts Dessert	450.00	100.000%	\$4.49	\$1.55	\$2.945	34.41%	\$2,059.42	\$707.60	\$1,349.76	LOW	HIGH	Workhorse
Average Item Profit: \$2.945 Menu Popularity Factor: 80.00%												

Figure 10: Sample Menu Engineering Worksheet⁵

Your tasks are completed.

Please answer the following questions with your immediate response, rather than thinking about items for a long time. All items should be answered. If you feel that you cannot respond to a particular item, please mark the centre point of the scale.

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
1. I think that I would like to use this system frequently	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2. I found the system unnecessarily complex	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3. I thought the system was easy to use	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4. I think that I would need the support of a technical person to be able to use this system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5. I found the various functions in this system were well integrated	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6. I thought there was too much inconsistency in this system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7. I would imagine that most people would learn to use this system very quickly	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8. I found the system very cumbersome to use	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9. I felt very confident using the system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10. I needed to learn a lot of things before I could get going with this system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

A.2 Questionnaire

Thank you for participating in this study. The purpose of this study is to understand how potential users interact with menuVis, a menu creating and editing tool.

In this study, you will to complete a task that involves interaction with menuVis. When you are working on the task, I'm going to ask you to think aloud, telling us what's going through your mind. Please tell us what you are doing, what you are trying to do, what you are thinking, what you find it works, and what you find it does not work. This will help us gain the most insight and improve the tool.

A participant number will be assigned to you. No personal data will be collected. Your interaction with menuVis will be screen recorded. We will also take notes during the test. The records will only be seen by people who are working on this study.

If you agree to participate in this study, please click the button below to start.



We thank you for your time spent taking this survey. Your response has been recorded.

Figure 11: The post-task survey given to users.

Task 1: Create a new recipe.

