Boba Shop Database Management Portal

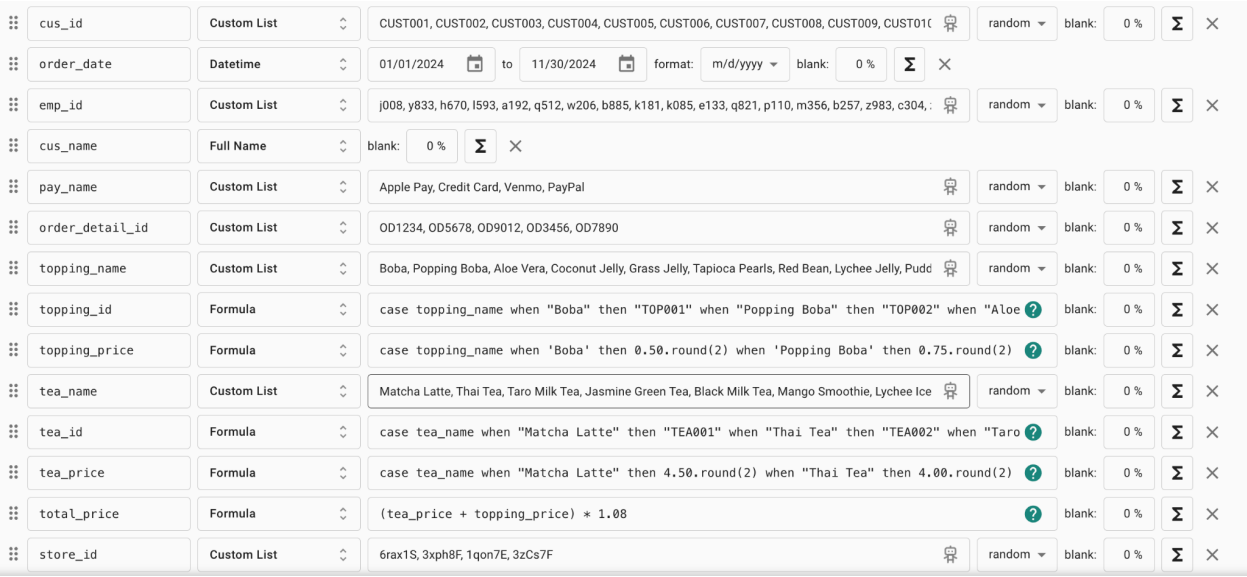
Boba 4 U: Yilin Sun, Alisha Kartik, Winston Chen, Yongxin Li

Introduction

Efficient selling and employee data management are crucial for the operational and analytical success of our starting business. This report details the design and functionality of a comprehensive system that leverages MySQL, Python, Qt Designer, and PyQt5 to streamline operations and generate actionable insights. The system utilizes a dataset containing selling and employee information, including customer orders, employee details, and sales analytics. Combining operational and analytical models is better for our inventory management, employee tracking, and data-driven decision-making.

Data Source

The primary data source for this project are generated manually from Mockaroo https://www.mockaroo.com/schemas/667973



Our menu has about 17 drinks and 10 toppings. This synthetic data reflects diverse customer preferences and purchasing behaviors.

Application Design:

The application contains 2 main models

Operational Module:

This module supports day-to-day activities such as order processing, and customer information.

- Order processing:
    - Employees order drinks and toppings for customers using an intuitive interface.
    - The system calculates the subtotal, applies about 8% sales tax, and displays the total price.
    - Orders can be added to a cart, and the preferred payment method (e.g. Apple Pay, Credit Card, Venmo, Paypal) is recorded.
- Order Details and customer information:
    - Captures order specifics, including data time of purchase, and the order id.
    - Stores customer details such as name and payment preferences to enhance analytical modeling.

Analytical Module

This module provides insights into sales trends, inventory needs, and employee data, supports strategic planning and operational efficiency through advanced reporting and data visualization:

- Top-Selling Products:
    - Displays the top 10 drinks and toppings based on sales volume.
    - Guides inventory preparation, ensuring popular items like boba are adequately stocked while implementing conditional preparation for sensitive items(e.g. chess foam is prepared only if fewer than 10 orders)
- Sales Reporting:

    - Generates daily, weekly, and monthly sales reports within a user-defined time range.
    - For example, selecting a time range from 06/24/2024 to 07/07/2024 allows the user to:
        - View daily sales totals for each day in the range.
        - Analyze weekly sales for two periods: the week of 06/24 and the week of 07/01.
        - Generate monthly bar charts for June and July.
- Employee Data Management:
    - Facilitates viewing employee details by selecting their name, and displays comprehensive information such as address, vehicle details, date of birth, etc.

Database Design:

The database of Boba Shop Management System has an operational module and analytical module.

Operational Module

The Boba Shop Management System uses a MySQL database to manage and store its data pertaining to the day-to-day operations of the boba shop. The database is designed to adhere to the normalization technique and all the tables are in 1NF. It consists of total 10 entities/tables viz customer, employee, employee phone number, vehicle, tea, topping, orders, order details, store and sells. These tables are designed to gather and organize all the data related to the basic operations of a boba shop. Further they also serve the purpose to investigate analytics of the business operations.

To design the database, the website ERDplus was used.

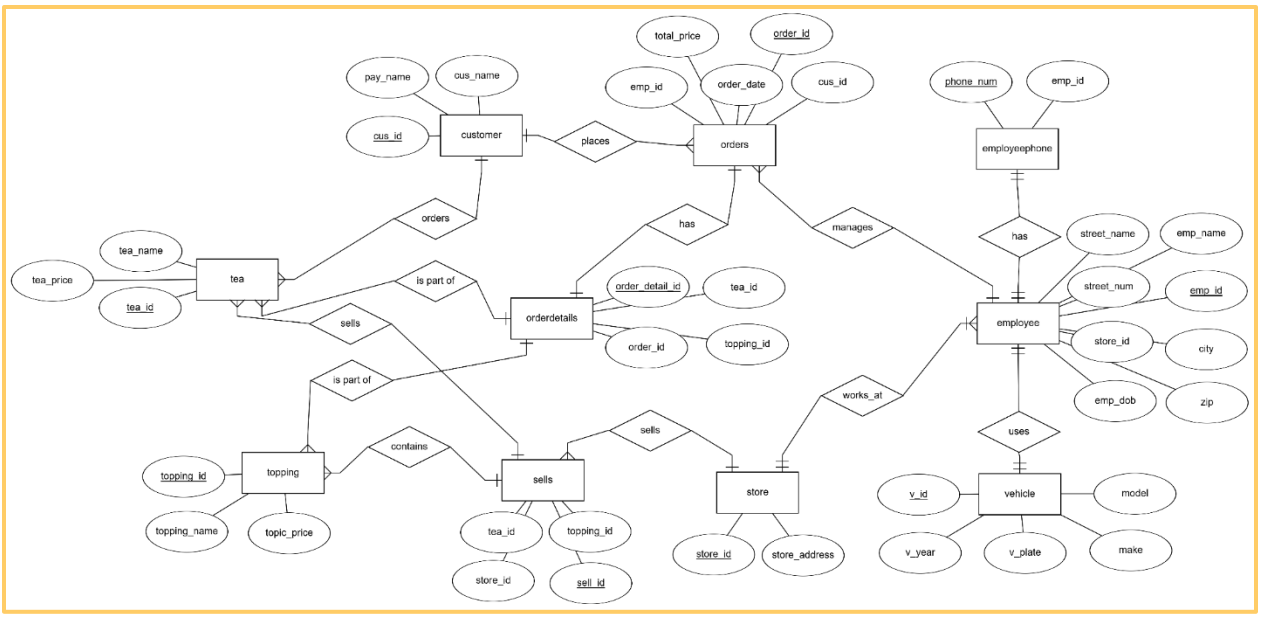


Figure: ER DIAGRAM

A comprehensive table explaining the figure- ER Diagram

Entity	**Attributes**	**Purpose**
Customer	cus_id, cus_name, pay_name	Stores customer information and their preferred payment methods.
Employee	emp_id, emp_name, emp_dob, store_id, city, street_name, streen_num, zip	Records information about employees and their assigned stores
Employee Phone Number	Phone_num, emp_id	Contains primary phone numbers of each employee
Vehicle	v_id, model, make, v_plate, v_year	Contains details of the vehicles owned by the employees
Tea	Tea_id, tea_name, tea_price	Contains details about available teas including prices
Topping	Topping_id, toppin_name, toppong_price	Contains details about available toppings including prices
Orders	Order_id, order_date, cus_id, emp_id, total_price, tea_id, topping_id	Tracks customer orders, including the responsible employee and total price.
Store	Store_id, store_address	Record of various stores and their location
Sells	Tea_id, toppin_id, store_id, sell_id	Tracks the tea and topping sold by various stores

The corresponding relational schema of figure: ER DIAGRAM is shown below:

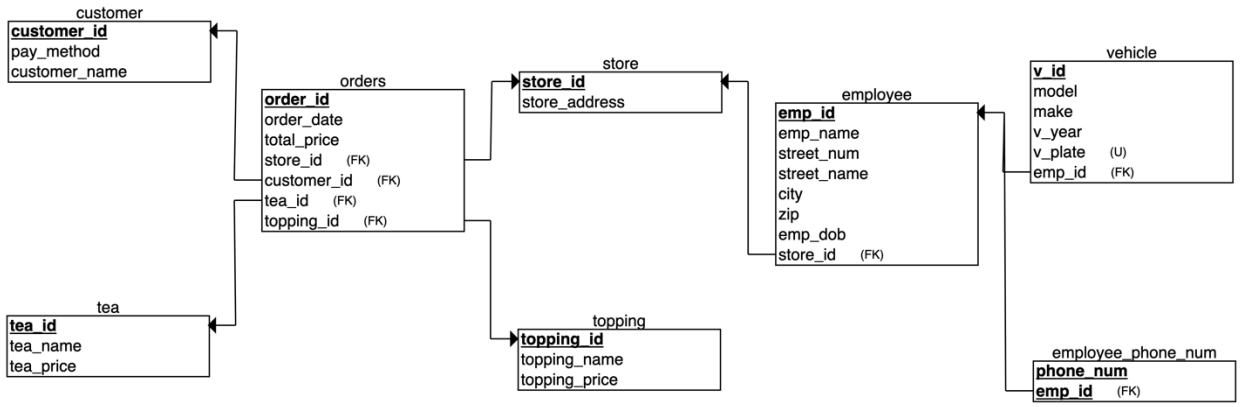


Figure: RELATIONAL SCHEMA

A-1 : Working of the operational module:

The landing page of the graphical user interface prompts user to choose among employee portal, manager portal and exit.

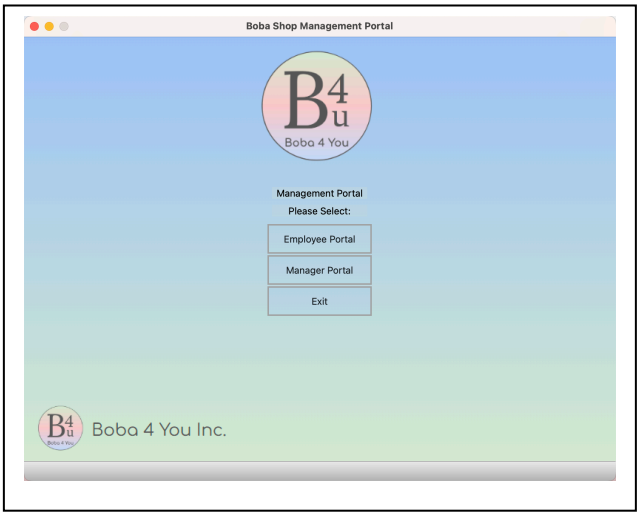


Figure:GUI- Landing Page of Application

Clicking on the employee portal takes us to the order entry page. Here we can take a new order from a walk-in customer by entering the required field.

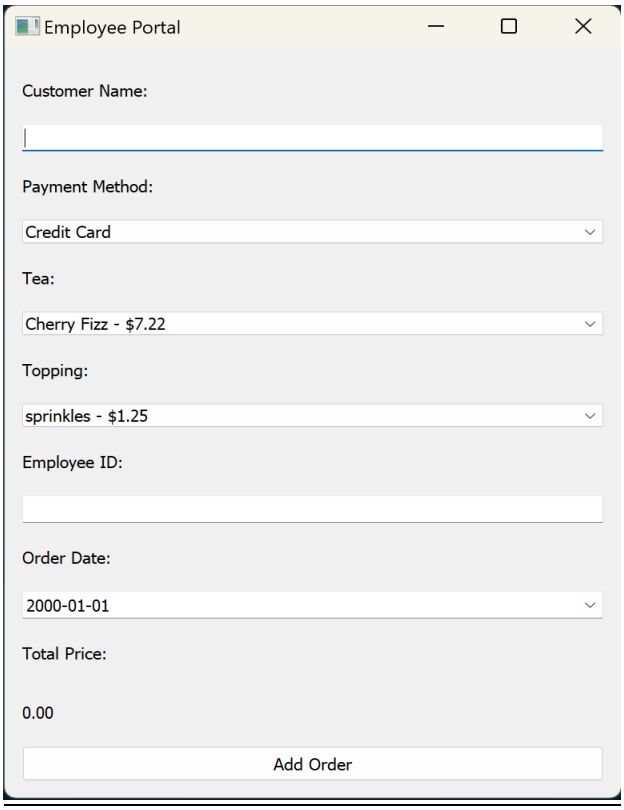


Figure: GUI- Employee portal

In summary, the employee portal is used to create a fresh order, and the corresponding entries are entered in their respective tables of the database. The total amount by adding the prices of tea and topping are displayed at the end before finalizing the order.

Analytical Module

The analytical module delivers insightful reports to review and aid in the business operations like sales reports, employee performance reports, most loved tea or topping.

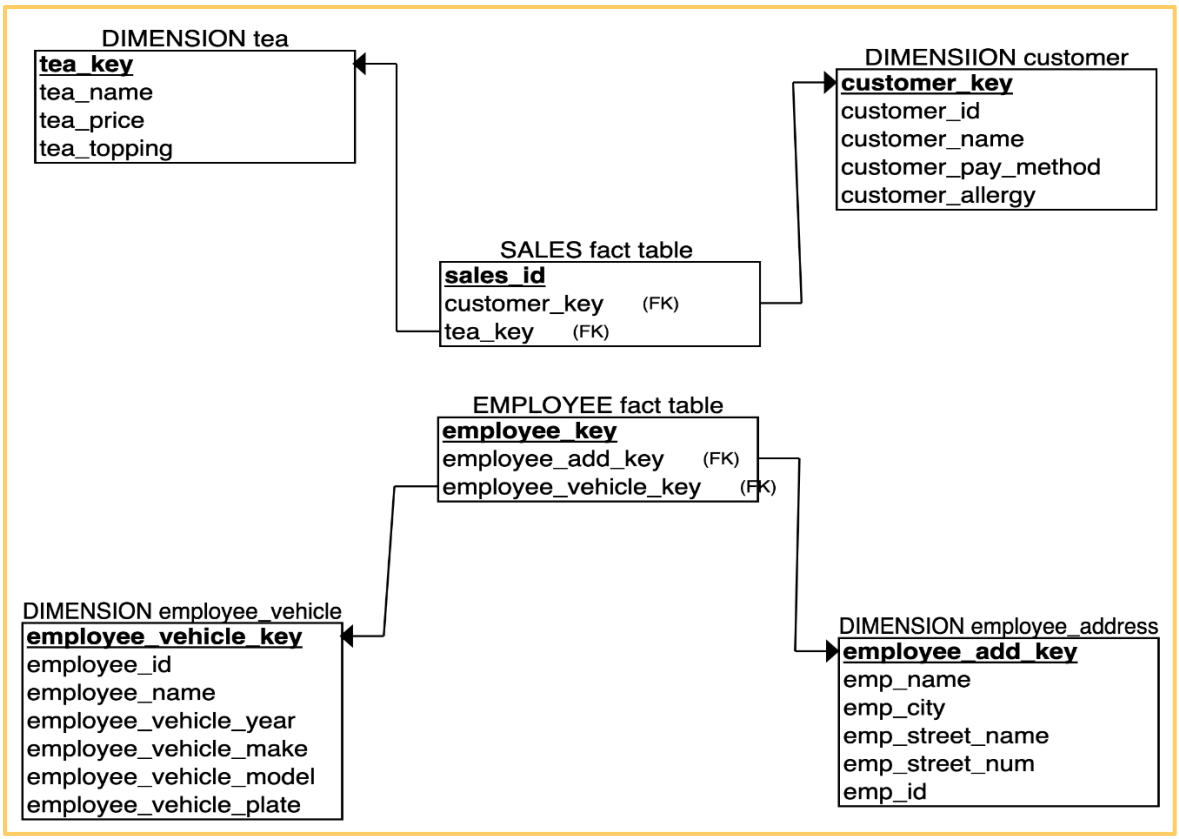


Figure: STAR SCHEMA

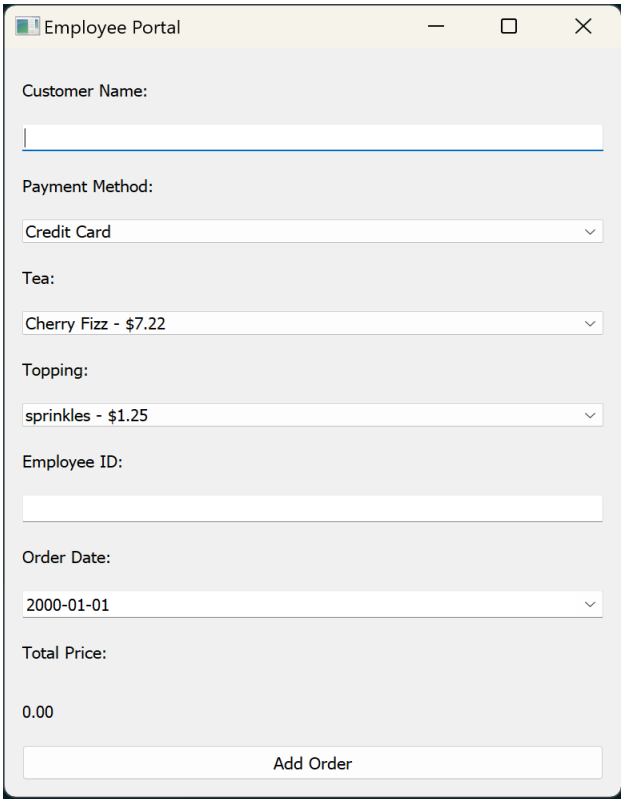
Working of the operational module

Operational Module Components

The employee portal is made from the employee_window.py file, which contains the UI made from QtWidgets as well as the queries needed when talking to the database server.

“Employee Order” Window

Once the “employee window” button is clicked, the employee will be taken into the orders page. This page contains all the necessary fields to take a customer’s order.



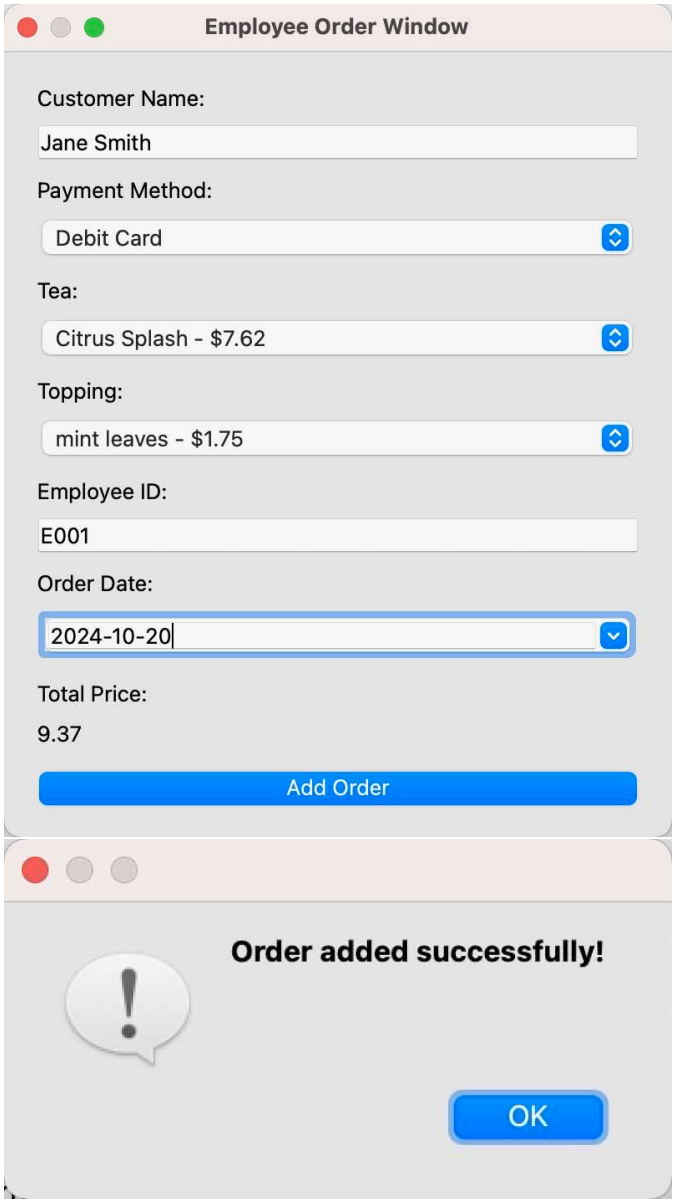
(default employee portal window)

Here the employee will take the customer’s name, order, and payment method, then input their employee ID and the order date to let the system recognize who authorized the transaction and when it occurred. Upon clicking the “Add Order” button, the order will then be sent to the orders table in the database where it can be used for analysis, labelled with an order_id. The drop down menus use SQL to query the available payment methods in the database and display

it in the text box when selected. The tea and topping drop-down menus query the appropriate tea and topping and is then formatted into a “tea name/topping name – price” pattern.

Example Use Case

Here is how the orders screen will look when filled, as well as the confirmation screen when the order is successfully sent back to the database.

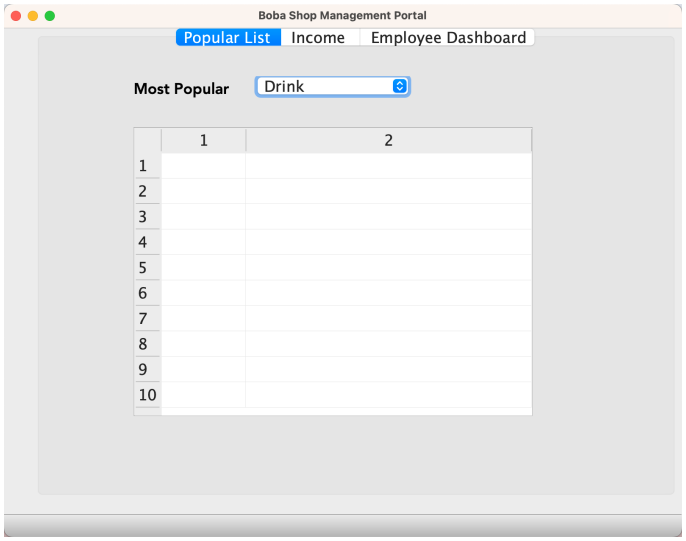


(example order with confirmation window)

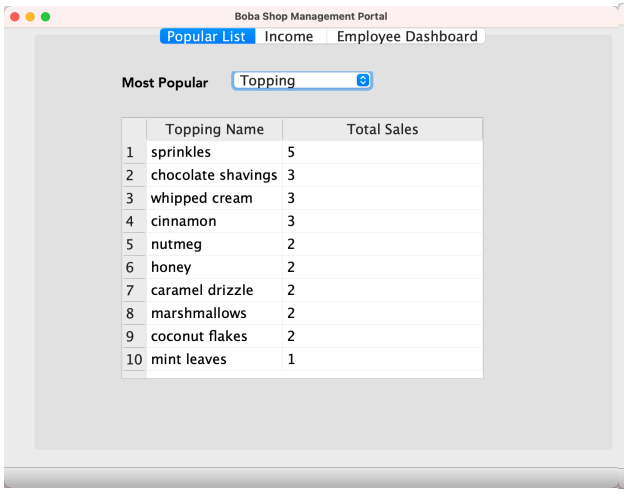
Working of the analytical module

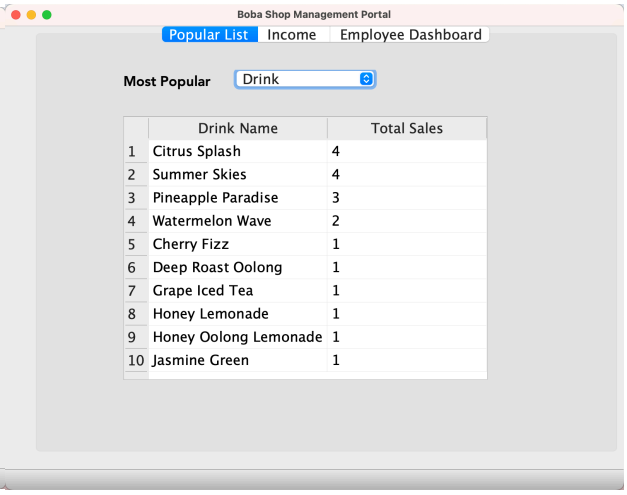
“Popular List” Window

As soon as the “Manager Window” button is clicked, the manager is taken into the analytical module of our boba shop management portal. The default window is the unpopulated “popular list” window.



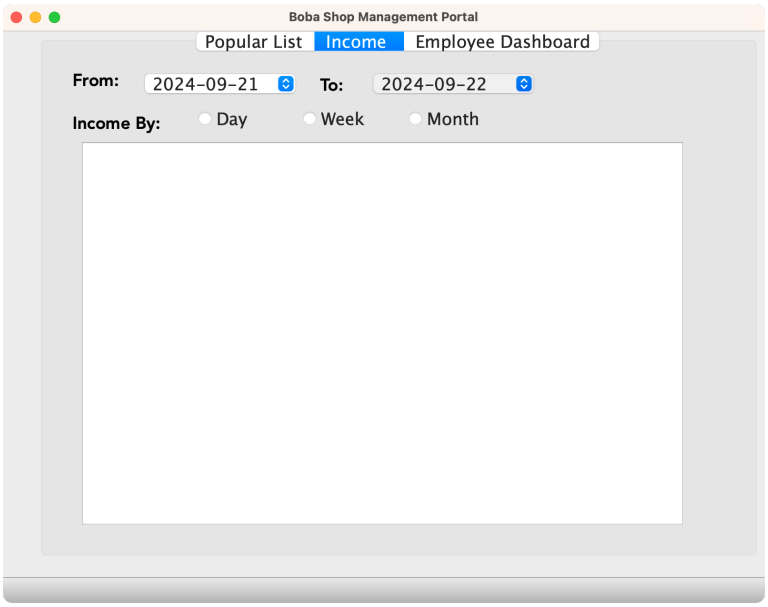
Here the manager can easily see the best-selling drinks or toppings from all our shops, as well as their historical sales count. Once we select the “toppings” option from the drop-down menu, we see the top-selling toppings; Once we select the “drink” option from the drop-down menu, we see the top-selling drinks. The column widths and names are adjusted accordingly.



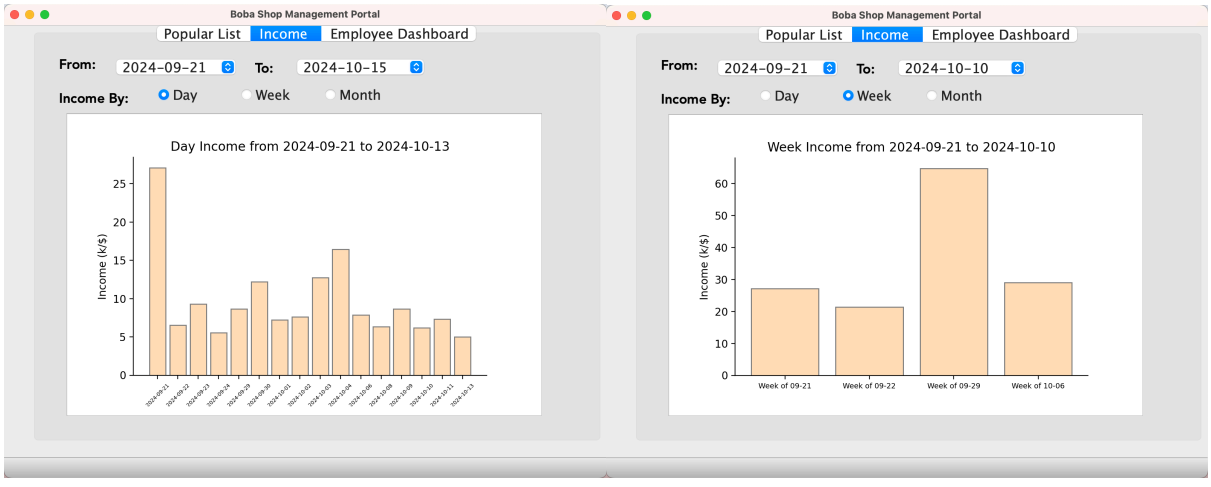


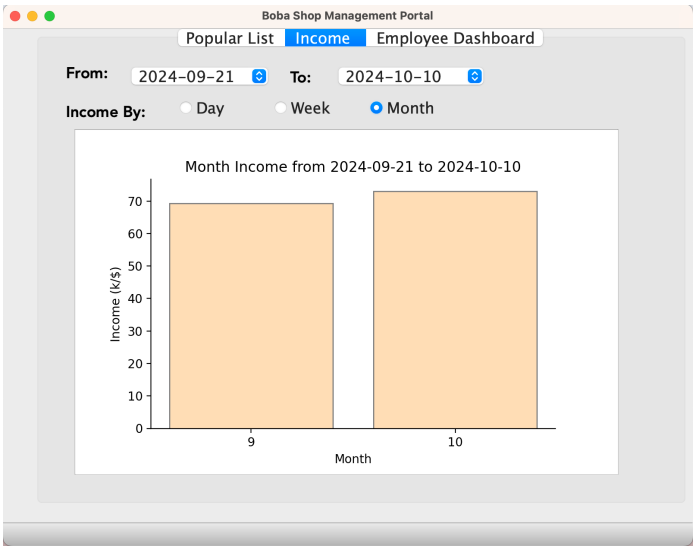
“Income” Window

The “income” window aims to provide a visual representation of the income of our boba shops over a selected time period. The manager can choose a time period based on the data we have collected, and choose to group the income by day, by week, or by month. Note that once the manager selects a date from the “From” drop-down menu, the “To” drop-down menu will change accordingly so only future dates will show up.



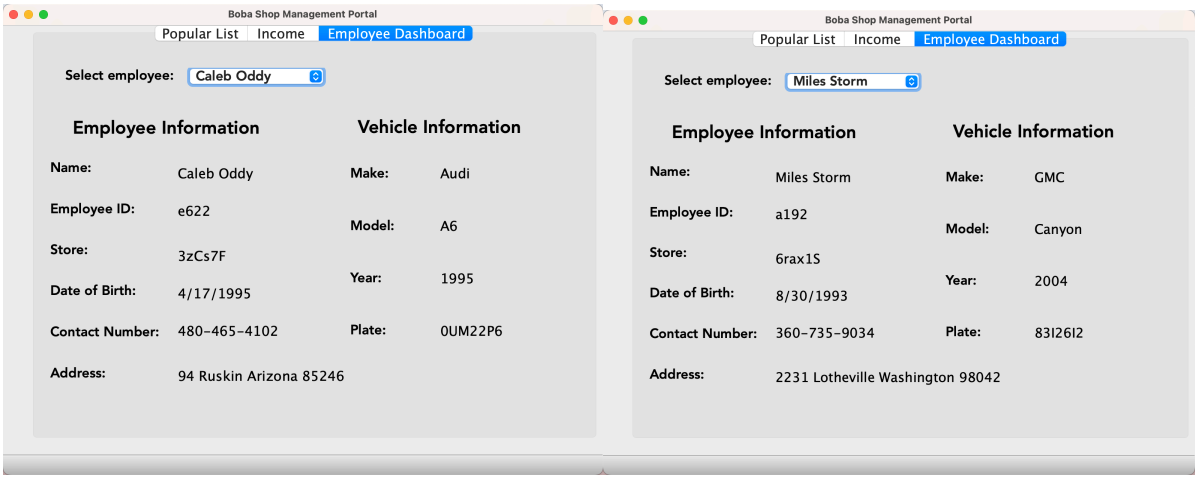
For any selected time period, the app will be capable of creating visualizations of summarized income by day, week, and month, as well as adjusting the x-ticks, title, and y-axis labels accordingly.





“Employee Dashboard” Window

This window aims to provide the manager some basic information of each employee, including their name, phone number, home address, date of birth, employee ID, and the store ID where they work at. In addition, there is also a dashboard on the employee’s vehicle information, including vehicle make, model, year, and plate number.



The manager can see all the current employees from the drop-down menu, and selecting an employee from the menu automatically populates the dashboard with employee information.