Technical Document

Members: Arjun Ray, Mohit Mahajan, Abhinav Garg, Parth Gera

Accomplishments

The web application accomplishes two goals of a restaurant operation.

- Enhances a Diners experience
- Provides business insights to the restaurants

The app workflow challenges the traditional ordering process in a restaurant and introduces a new system design where customers could order through a digital menu and track their orders through a web interface. Further, as a consequence of the new design, restaurants become capable of capturing the data about the operational aspects of the business. The insights generated on those data points, ultimately help the restaurant/restaurant managers to have a greater visibility of the operations and improve the business accordingly.

Usefulness of the project: The Real-World Problem

Waiting is frustrating while dining out. We all have been in a situation while dining out when we're waiting for someone to take our order and then once it is punched, we don't know if the order is ready or not. We've also certainly experienced that sometimes the order is missed out then we need to remind the waiter to look into our order and get the food faster. It's not a good experience for the customers neither the restaurants, and it occurs because of the inefficiency in the catering system. Solving that problem is exactly the intention of the app, our objective was to develop a catering application which can be used at every table (in form of kiosks/booths/or phone app) in restaurants, food joints, clubs, bars, etc. This app would have two kinds of user personas.

This would be used by customers to look at the menu, place orders, and pay bills. The application would provide a status of available items on the menu basis the raw material inventory available in the restaurant. The customers would be able to add items in their cart and place orders and then pay bills being at their table. This way the application would enable no-contact ordering that will reduce order-queue wait times and make the ordering process hassle-free. This would also be preferable during the unprecedented times of social distancing which brought a major shift in the ways of communication and physical interaction in places of gathering.

On the other side, this would assist restaurants to replace the pen-paper process with a systematic order management process. The order placed by customers would enter a queue consisting of all orders and be assigned basis priority/preparation time. Once the order is ready, it'll be available at checkout for customers to pick up. At the end of their meal, customers can drop reviews under their order for future improvements. That way will have greater visibility into the catering operations.

The data captured by the system can be broadly divided into 4 segments:

1. Orders
2. Customers

3. Inventory
4. Payments

It can be further used for generating insights that would help restaurants make better operational decisions regarding promotions and procurement. A few analytics applications could potentially be:

1. MoM raw material forecasting based on orders
2. Daily customer trends
3. Capacity/resource planning in the restaurants based on hourly demand
4. Sentiment analysis based on rating and food reviews

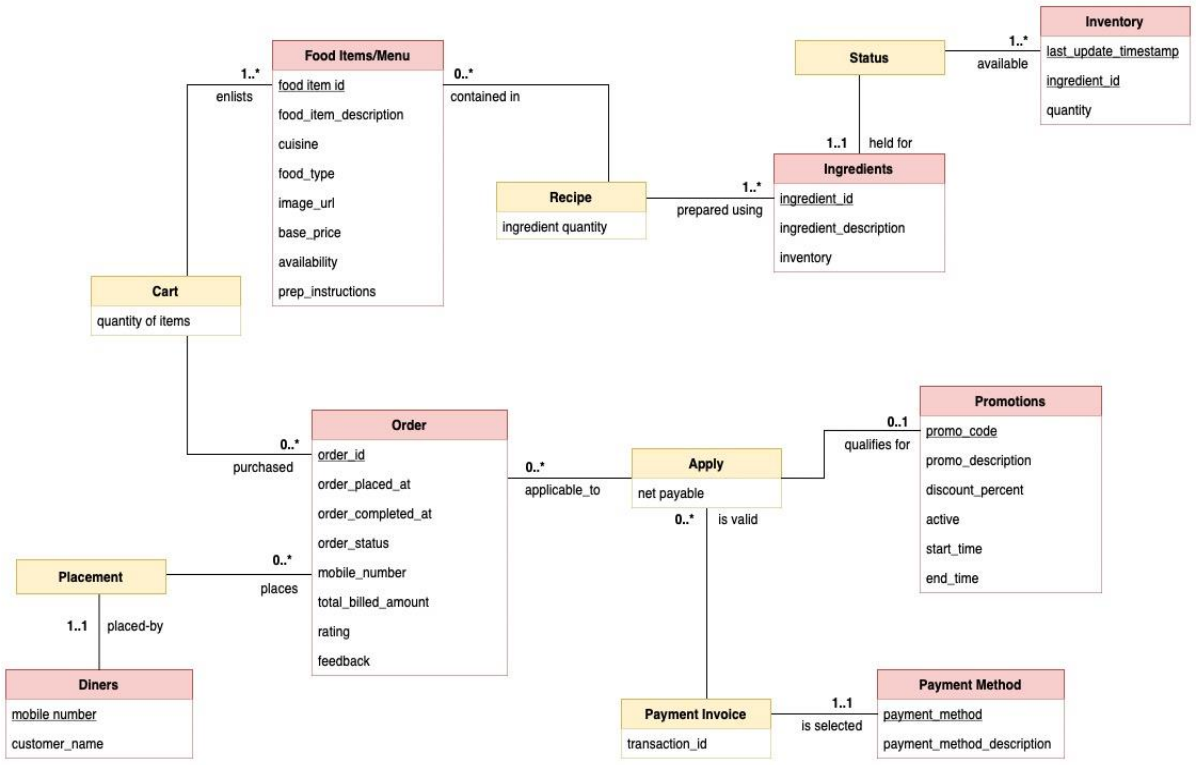
The software system is scalable and with the improved accessibility of web and cloud technology, it could be affordable and used by small and medium restaurants and help them to streamline their catering operations. Using the business tools of the app as described, the restaurant managers would be able to run the business more efficiently as they could make informed decisions based on the data. Beyond the usual efficiency improvements, such a system would also be effective in a modern (post-covid era) setting where people want to reduce the spread of infections by eliminating the contacts. Due to very same reasons, most of the bars and restaurants have already adopted the QR code-based menu which we think is a good enough market validation for the success of such a system, we intend to more capabilities to the archaic “pdf menu” to be a radical improvement in the customer experience.

Data in our database

1. Menu of items the restaurant serves
2. Recipes (list of ingredients) needed for each menu item
3. Inventory of ingredients the restaurant has
4. Table of orders that customers have placed
5. Details of items purchased in each order and the corresponding payment method
6. Personal details of diners to communicate about promotions and offers

ER/UML Diagram

Please check the following page



Database schema (DDLs) and index design analysis.

The database DDL has been put together in the sql script on GitHub
Note: The app has been developed in Django python. This script is run when Django server deploys the app. The indexing analysis was done for several complex operations. The results of indexing analysis have been published here

Data Collection

Data required for the application instance was obtained by combining two data sources. On one side we needed the data for demand which could be used to further extrapolate the data for inventory and orders. On the other side, we needed the data for the food items, menu and the recipes to map inventory to the demand. To replicate the data for demand we assume that the offline and the online ordering patterns are similar and therefore the demand data was obtained through one of the Kaggle sources Link .

The data for menu and recipes was obtained through another Kaggle dataset Link

The two dataset were connected by mapping the food items in one with the another and he ingredient and recipe data was generated by properly parsing the recipe column in the food items data (2).

We used an intelligent extrapolation mechanism to generate the inventory date. We assumed a material inflow and outflow for each week and based on the material flow we could generate the inventory data

for ingredients.

```
Inventory Available (t) = Inventory Available (t-1) + Inflow (t) –
Outflow (t-1)
```

Our data generator scripts have been packaged in the repository here.

Application design and the features involved

Our application has two interfaces:

1. Customer Interface:
    - Customers could look at the menu of the restaurant in the digital form
    - Customers could search for food items in the menu
    - Customers could select several items from the menu to order
    - Before placing the order, customers could edit the order
    - Customers have the option to pay using several payment methods
    - Once the order is placed, customers could track the order through their mobile number
    - Once the order has been completed, customers could use their mobile number to retrieve the order and provide rating/feedback to the restaurant.

2. Restaurant Interface:
    - The employees at the restaurant can login using the option provided
    - Restaurant have a kitchen view where they could access all the open orders (orders that haven't been served yet)
    - Restaurant employees could access the historical orders and retrieve demand pattern and revenue earned for different time horizons rolled up for different time periods using interactive visualization
    - Restaurant employees can view the historical order details
    - Restaurant employees can view the inventory on hand and plot of inventory status through time through an interactive visualization
    - Restaurant employees can update inventory based on a new purchase (inflow of the materials)
    - Restaurant employees can add/edit/delete menu items, recipes and ingredients data
    - Restaurant employees can look at several business insights to improve customer experience, determine staffing needs and promotions

Besides the user interfaces the application has couple of internal features in the middleware.

1. Based on the customer order placed for some food item, the system would determine the ingredients required and accordingly update the inventory of the ingredient so that the restaurant employees can view the real time inventory.
2. Based on the time when order is placed, if any promotion is active during the order placement, the system would automatically apply the promotion and let the user know about the details, this is a cheery on top!

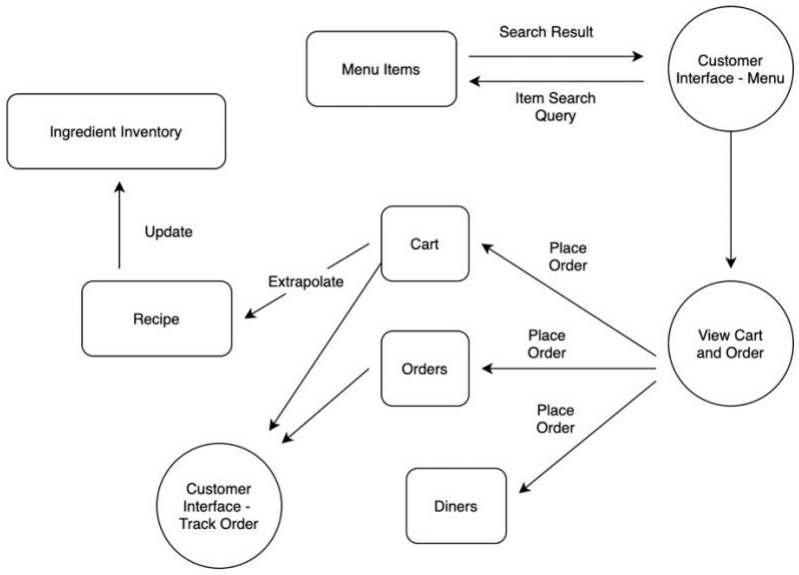
The middleware features are implemented through an advanced database program, the details for the same are covered in the following section.

Advanced Database Program

We have chosen stored procedure + trigger for our advanced database program. The sql code for the stored procedure can be found here and trigger can be found here. Each menu item requires some raw ingredients according to its recipe for preparation. Whenever a user orders any food item, our stored procedure looks up its recipe and calculates the quantity required for each raw ingredient to cook that food item. After that, it automatically inserts new entries in the inventory table by calculating the updated quantity available for all those ingredients at the latest snapshot. This gives restaurant a real-time view of its available inventory and they don't need to physically look it up. Storing the inventory in this way also helps to capture the inventory movement and can help restaurant plan better. For example, restaurant can estimate when the inventory of a particular food item will stock out based on the available inventory and its trend, and can then plan to restock accordingly.

Our trigger automatically applies promotion to the final bill. Sometimes, it may happen that users are not aware of any running promotion when they checkout their order. Our trigger looks up any active (and applicable) promotion and updates the final bill of the customer. The restaurant also doesn't have to worry about manually updating the bill with promotions. They only need to maintain the promotions database with timely updates.

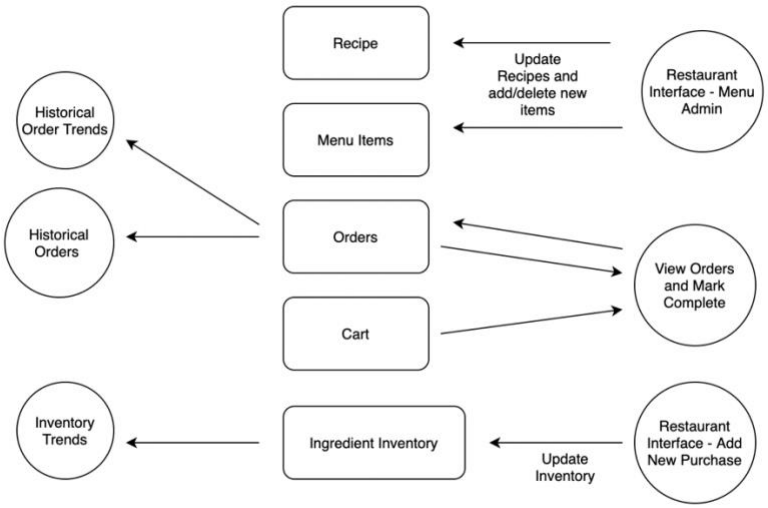
Data Flow:



The above diagram shows the data flow behind the customer interface.

The application has a menu where the customer can search for an item (e.g. Amritsari Kulcha, Dal Tadka etc.). This sends a search query to the menu relation and returns the relevant results. The customer can then select an item and quantity that is to be ordered. Once selected, the customer can view the card

and submit the order. Placing an order triggers data insert into relations Cart, Orders, and Diners. The application provides the customer to track the order. In parallel, the items selected in the cart are mapped to the recipe relation in order to get the ingredients required. These ingredients with the quantity required are reduced from the remaining quantity in the inventory relation.

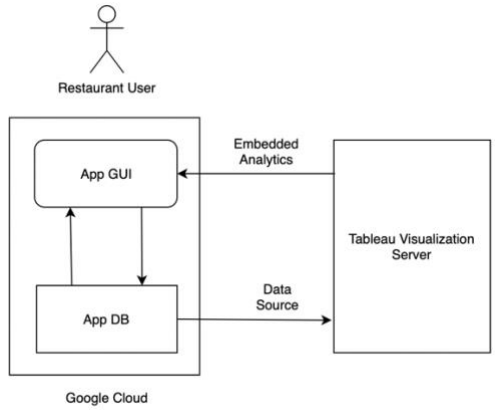


The above diagram shows the data flow behind the restaurant interface.

The application has a menu admin screen where updates (add/delete) related to recipes can be made. These updates are made to the recipe and menu relations. The restaurant manager can view pending/completed orders as well as mark orders as completed when the order is delivered. These changes are made in the Orders and Cart relation. The advanced queries related to the highest ordering sellers as well as the WoW/MoM trend of orders are executed at this stage to draw insights and make decisions. These insights are visualized using Tableau dashboards embedded in the application. The restaurant also has the interface to update inventory of ingredients as in when they are replenished. This change reflects down to the inventory relation and is also used understand inventory trends to make improvements in inventory replenishment decisions.

Technical Challenge and Learning

We wanted to have an advanced component of interactive visualization in order to make our app feature rich and help the restaurant owers realize the importance of data analytics in business. In our vision, this feature should also make the restaurants capable of quantifying the impact of decisions they take in their operations. To implement this for the purpose we needed a visualization that could provide us capability to drill down and roll-up the demand, revenue and inventory data at different levels of time horizon. Within the limited scope, skillset and time we needed something that could be a quick turnaround. Tableau helped us here. Tableau is an interactive visualization platform that could connect to the app database and has the BI capabilities of plotting besides aggregating and dis-aggregating data at different level of details. Further, there's an interesting feature in tableau that a user could host the dashboards on the web on a dedicated instance. The dashboards are rendered as html iframes in the web which have the option to be embedded into another webpage. We thought that it was certainly a right fit in our context. We connected tableau to our app database to build a dashboard then hosted this dashboard on tableau server (bought a 15 day free trial plan) The hosted dashboard was then embedded in our app as an html iframe component to bring pull together the interactive visualization component.



The restaurant admin and user make updates to the database instance. Tableau server fetches the live data from the database instance and renders the dashboard, the dashboard embedded in the app provides the interactive visualization capabilities. We think, this way we could quick build professional grade visualizations and interactivity features in the app and could be used in situations with limited time, scope and skillset. It may cost some money but this would delight the users.

Project Management

In the original specifications, we planned to integrate forecasting of ingredient inventory to determine appropriate replenishment. In the last few weeks of the development, we were already working to make the components of the app stable and make sure they do not crash due to unknown errors and hosting the app on GCP took more than expected time for deployment.

We already had developed the function for forecast in the backend, but we were left with no time to

develop a GUI for the same. Therefore, we do not mention we development of this piece in our presentation. We plan to integrate this piece later after the course completion. The final division of labor is mentioned below:

Team Member	Assignment
Arjun Ray	Web App Development (Django), Interface design, Database Implementation
Mohit Mahajan	Database design, Interactive Visualization, GCP hosting, Data Generation, Database implementation
Parth Gera	Advanced Queries, Data Generation, Stored Procedures, Triggers
Abhinav Garg	Stored Procedures, GCP Hosting, Data Generation, Database design