COS214 PRACTICAL 5 2023

Restaurant Simulator

University of Pretoria

Computer Science

COS214 Practical 5 2023 – Restaurant

Group Members

* **Aiden Govender - u22520458**
* **Mishka Dukhanti - u22617541**
* **Rethakgetse Manaka – u22491032**
* **Victor Igbojinna - u22492616**
* **Kamogelo Moeketse - u22623478**

**Introduction**

Our team is excited to present a proposal for the design of a restaurant simulator, which can also be adapted into a "restaurant tycoon" game. This simulator/game aims to capture the essence of a bustling restaurant environment, where various processes seamlessly come together to produce and serve food to customers. While we understand that there are multiple ways to approach this project, we will focus on two primary areas: the restaurant floor and the kitchen, exploring how these components must communicate to create a realistic restaurant experience.

**Project Overview**

Our restaurant simulator/game will provide players with an immersive experience, giving them a taste of the challenges and excitement that come with managing a restaurant. This project will not require multithreading, ensuring simplicity and accessibility for a wide range of users.

**Requirements**

**Floor Requirements**

1. **Table Assignment**
   1. The system should allow customers to request to be seated.
   2. The Maître D or hostess should allocate tables if available.
2. **Order Placement**
   1. Customers should be able to place orders.
   2. Waiters should be able to take orders when requested.
   3. Customers should have the option to delay ordering if they are not ready.
3. **Order Communication**
   1. The system must facilitate the transfer of customer orders to the kitchen.
   2. Waiters should pass orders to the kitchen staff/mediator.
   3. The kitchen should prepare orders efficiently and inform the waitstaff when orders are ready.
4. **Table Management**
   1. Waitstaff should manage tables, including clearing and cleaning them for new customers.
   2. Tables should be combinable or split as needed to accommodate different party sizes.
   3. The system should ensure that customers are seated at appropriate tables based on their party size.
5. **Payment and Billing**
   1. Generate and present bills at the end of the meal.
   2. Allow customers to request splitting the bill among various parties at the same table.
   3. Optionally, provide the ability for customers to start a tab for later payment.
6. **Waiter Assignment**
   1. Each waiter should have a set of assigned tables for which they are responsible.
   2. Waiters should be allocated tables in a way that optimizes service.
7. **Customer Expectation Management**
   1. Implement a system to track customer satisfaction.
   2. Manage customer expectations by ensuring timely service, order accuracy, and quality.
   3. Handle customer complaints and feedback effectively.
8. **Customized Order**
   1. Allow customers to customize their orders from a list of available options.
   2. Support special requests, such as specifying preparation methods(e.g., grilled vs fried).
9. **Booking System**
   1. Integrate walk-in customers into the seating process.
   2. Implement a booking system to manage reservations, if possible.
10. **Tipping System**
    1. Calculate tips based on customer satisfaction and bill amount.
    2. Provide a mechanism for customers to leave tips, if possible.

**Kitchen Requirements**

1. **Order Reception**
   1. The kitchen should receive orders from the mediator.
   2. Orders should be processed in the order they are received, following a first-come, first-served approach.
2. **Order Preparation**
   1. Different chefs or kitchen staff should be responsible for various parts of the preparation process.
   2. Each staff member should focus on a specific task, such as grilling or frying.
   3. Chefs must follow the recipe and preparation methods accurately.
3. **Order Routing**
   1. A dish may need to be passed between different stations within the kitchen before returning to the head chef for final plating.
   2. The system should manage efficient routing of dishes to the appropriate stations.
4. **Completion Notification**
   1. Once an order is completed, the kitchen staff should notify the mediator which notifies the waiter.
   2. Waiters should be informed that the order is ready for pickup and delivery to the customer.
5. **Order Queue Management**
   1. The system should maintain a queue of orders in the kitchen.
   2. Orders should move through the queue as they are prepared, ensuring a smooth workflow.
6. **Inventory Management**
   1. Monitor the availability of ingredients and kitchen equipment.
   2. Alert kitchen staff or management when supplies are running low to avoid delays in food preparation.
7. **Additional Considerations**
   1. Maintain a log of order progress and any issues or delays in the kitchen.
   2. Support ability to adapt to changes in the order queue based on customer requests or special circumstances.

**Additional Requirements**

1. **Bars with Various Cocktails**
   1. Ensure that the bar can serve all drinks offered on the menu. Alcoholic or non-alcoholic.
   2. Implement a system where drinks can be delivered independent of the food that they ordered with the drink.
2. **Valet Service**
   1. Create a mechanism to offer valet service to customers.
   2. Manage valet staff, parking areas and retrieval of customers’ vehicles.
3. **Inventory and Accounting**
   1. Keep track of expenditures, revenues, and profits.
   2. Integrate an accounting system to handle bills, payments, and payroll.
   3. Integrate a tab system to keep track of customers’ outstanding debt.

**Design Patterns Identified**

1. **Decorator**
   1. In the kitchen the, the decorator pattern can be used to dynamically add toppings, sauces, or additional ingredients to a base food item.
   2. Each decorator class represents a specific modification that can be added to the base food, allowing for flexible and extensible customization.
2. **Prototype**
   1. The prototype can be used applied to efficiently create plates or dishes in the kitchen.
   2. Instead of creating new plates from scratch, the kitchen can clone prototype plates, reducing the overhead of plate creation and preparation for each order.
3. **Abstract Factory Method**
   1. The abstract factory method pattern can be used to create different types of chefs, each specializing in preparing a category of dishes**.**
   2. Each concrete factory corresponds to a specific type of chef, and they produce chef objects with different skills and responsibilities.
4. **Factory Method** - Different types of drinks and the valet service.
   1. In the bar area, the factory method pattern can be used to create different types of drinks (e.g., cocktails, non-alcoholic beverages).
   2. With the valet, the factory method pattern can be used to create different valet services.
5. **Mediator**
   1. The mediator design pattern can facilitate communication between the kitchen and the waiters.
   2. The mediator acts as an intermediary that centralizes and manages communication between the floor and the kitchen. Ensuring that orders are relayed, and order status is updated efficiently.
6. **Chain of Responsibility** 
   1. The chain of responsibility pattern will be used to handle various customer requests and complaints in a structured manner.
   2. Each Handler(e.g. waiter, handler) in the chain processes specific customer requests, such as seating, ordering, or addressing complaints, before passing it along the chain.
7. **Builder**
   1. The builder pattern can applied for constructing the actual orders with customized food orders.
8. **Visitor** – Manager
   1. The visitor pattern can be used to implement a manager who visits tables and interacts with customers.
   2. The manager can inspect customer satisfaction, handle special requests, and provide assistance, acting as a visitor that visits different customer tables.
9. **Observer**
   1. The observer pattern can be used to keep track of the inventory in the restaurant and the inventory automatically gets updated as to how much stock is left.
10. **Composite**
    1. The composite pattern can be applied to represent bills as a hierarchy of components, where the total bill is composed of individual customer bills where payment is split or complete bill where payment is combined.
    2. This pattern enables the construction and calculation of the final bill by traversing the composite structure.
11. **State**
    1. The state pattern can be used to manage the state of the restaurant tables regarding availability and occupancy.
    2. Tables can transition between states, such as “vacant”,” occupied”, or “reserved”, based on customer interactions, reservations, and dining progress.
12. **Memento**
    1. The memento pattern can be applied to capture the state of orders, bills, or accounting data at specific points in time.
    2. This allows for order preservation, auditing, and the ability to revert to previous states in case of errors or disputes.

**Keep things that belong together, together.**

**Testing**

**Github actions**

**More than fewer**

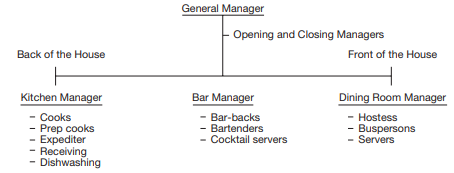
**Plan out the demonstration**

**Talk about patterns**

**Explain each pattern**

**Restaurant Operations  
The Floor**Restaurant operations are generally divided between what is commonly called front of the house and back of the house. The front of the house includes anyone with guest contact, from the hostess to the bus person. The general manager or restaurant manager runs the restaurant. Depending on the size and sales volume of the restaurant, there may be more managers with special responsibilities, such as kitchen manager, bar manager, and dining room manager. These managers are usually cross trained in order to relieve each other.

Most restaurants customer handling begins the moment the customer arrives with services such as valet services thereafter the customer is handled at the door by “greeting staff” or in our case a Maitre’D. Customer handling happens from these moments all the way to the bathroom up until they leave.

Below is the general hierarchy of the managers involved in the operations of the restaurant, both the front of the house as well as the back of the house.  


Often, when taking orders, the server begins at a designated point and takes the orders clockwise from that point. In this way, the server will automatically know which person is having a particular dish. When the entrees are ready, the server brings them to the table. He or she checks a few minutes later to see if everything is to the guests’ liking and perhaps asks if they would like another beverage. Some restaurants, servers are allocated a certain number of tables, which may vary depending on the size of the tables and the volume of the restaurant. Normally, five is the maximum. In other restaurants, servers rotate within their section to cover three or four tables. Servers introduce themselves and offer a variety of beverages and/or specials or invite guests to select from the menu. When taking the orders, the server will automatically know which person is having which dish. When the dishes are ready the server will bring them to the table. He or she checks a few minutes later to see if everything is to the guests’ liking and perhaps asks if they would like another beverage.

**Suggested Steps to service a table of customers:**

1. Greet the guests.
2. Introduce and suggestively sell beverages.
3. Suggests appetizers.
4. Take orders.
5. Check to see that everything is to the guests’ liking within two bites of the entrees.
6. Ask if the guests would like another drink.
7. Bring out dessert tray and suggest after-dinner drinks and coffee.

In our case we would have to include how the customer decides to pay. Will they pay immediately or will they open a tab and decide to add their bill to the tab. As well as we need to account for whether the bill will be a split bill or a combined bill.

**The Kitchen**The back of the house is generally run by the kitchen manager and the term refers to all the areas with which guests do not normally come in contact. This includes purchasing, receiving, storing/issuing, food production, stewarding, budgeting, accounting, and control.One of the most important aspects to running a successful restaurant is having a strong back-of-the-house operation, particularly in the kitchen. The kitchen is the backbone of every full-service restaurant, so it must be well managed and organized. The kitchen manager checks the head line-cook’s order, which will bring the prep (preparation) area up to the par stock of prepared items. Most full-service restaurants have similar layouts and designs for their kitchens. The layout consists of the receiving area, walk-ins, the freezer, dry storage, prep line, salad bar, cooking line, expediter, dessert station, and service bar area. The cooking line is the most important part of the kitchen layout. It might consist of a broiler station, pickup area, fry station, salad station, sauté station, and pizza station. Our main focus when designing the system will be on the Chef de partie.

A chef de partie or “station chef” and sometimes also called “line cook” is in charge of a particular area of production. In large establishments, chef de partie have assistants and cooks, but in most normal level restaurants, the chef de partie is the only worker in that department. These line cooks are again divided into — “first cook,” “second cook” and so on.

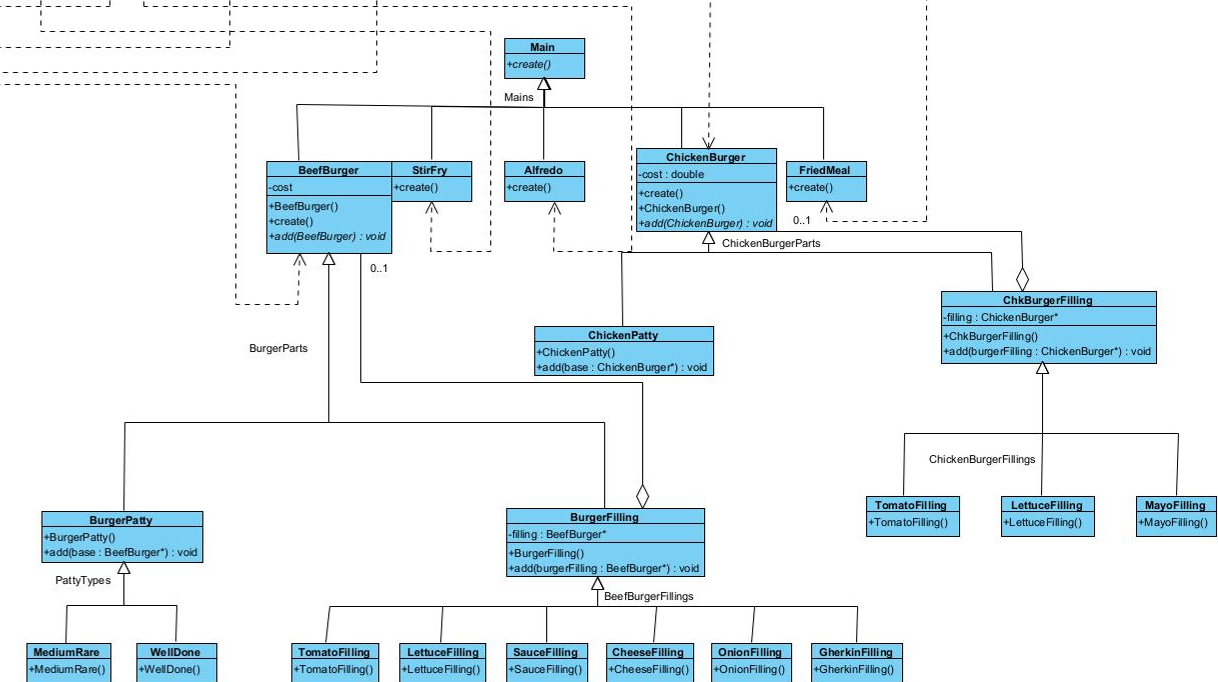
Types of Chef de Partie:

* Butcher-Chef
* Fish Chef
* Roast Chef
* Grill Chef
* Fry Chef
* Soup Chef
* Vegetable Chef
* Pantry Chef
* Pastry Chef

Chefs who handle orders as well as cook will create an inefficient process in the food production process. Thus, many large and high-end restaurants employ the use of a head chef to be responsible of the food production process. They are responsible for handling incoming orders and notifying the correct station about this order. Note that this is not their only responsibility. They are also responsible for ensuring that the process delivers food of high standards on time to the customer.

Practicing proper staffing is absolutely crucial for the successful running of a kitchen. Crucial to the smooth running of the kitchen is having a competent staff. This means putting the best cooks in the appropriate stations on the line, which will assist in the speed of service, the food quality, and the quality of the operations.

**How did we achieve our goal?**Using the patterns we identified this is how we applied them to produce a solution to the problems described in the brief.

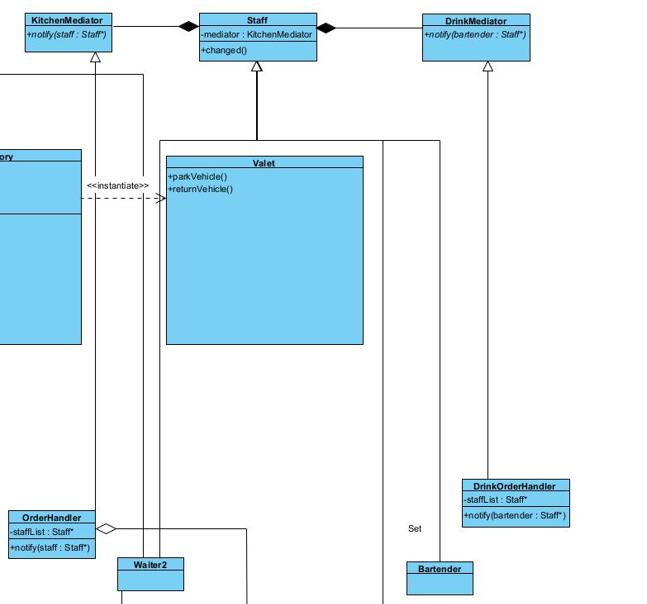


1) Decorator Pattern - Decoration of burger/food:  
Problem: The kitchen needed to dynamically add toppings, sauces, or additional ingredients to a base food item.  
Solution: We used the decorator pattern to create decorator classes representing specific modifications that can be added to the base food, allowing for flexible and extensible customization.  
We used this pattern to because we believe that it would allow us the flexibility we need for our food items on the menu.

2) Builder Design Pattern - Generation of plate in the kitchen:  
Problem: The kitchen needed to be able to generate a plate with different ordered items for each order which can become a complex ordeal.  
Solution: We applied the Builder design pattern to create plates with different food items thus allowing the kitchen to separate the construction of a plate to different chefs.  
We used this pattern because we realized that it would help easily create a “plate” for the customer and this pattern is perfect for us because it allows us to add many different food items together onto one “plate”  
  
A diagram of a computer

Description automatically generated

3) Abstract Factory Method - Different types of chefs:  
Problem: The kitchen needed to create different types of chefs, each specializing in preparing a category of dishes.  
Solution: We implemented the abstract factory method pattern to create different types of chefs using concrete factories, each producing chef objects with different skills and responsibilities.  
With our research we realised that there are many different types of chefs and this pattern allows us to have different chefs which create different food items.

4) Factory Method - Different types of drinks and the valet service:  
Problem: In the bar area and valet service, we needed to create different types of drinks and valet services.  
Solution: We used the factory method pattern to create different types of drinks and valet services, allowing for extensibility and customization of these services.  
Why did we use these patterns Ask aiden.  
  


5) Mediator Pattern - Communication between kitchen and Waiters:  
Problem: Our system needs to efficiently manage communication between the kitchen and waiters, ensuring orders are relayed and order status is updated.

Solution: We applied the mediator design pattern to centralize and manage communication between the floor and the kitchen, improving order management and status updates.

Our original idea was to pass the order from the waiter directly to the chef. We realized the use of the mediator pattern would allows easy communication between the waiters and the chefs with regards to communication about the orders.

A diagram of a server

Description automatically generated

6) Chain of Responsibility Pattern - Customer Handling:  
Problem: Our system needs to handle various customer requests and complaints in a structured manner.

Solution: We used the chain of responsibility pattern to process specific customer requests, such as seating, ordering, or addressing complaints, in a structured manner.

7) Visitor Pattern - Manager:  
Problem: We need our system to allow interaction with customers, inspect satisfaction, handle special requests.

Solution: We used the visitor pattern to implement a manager who can visit tables and interact with customers, addressing their needs and providing assistance.

8) Observer Pattern - Keeping track of inventory:  
Problem: Our system needs to keep track of restaurant inventory and update it automatically.

Solution: We implemented the observer pattern to automatically keep track of inventory changes, ensuring accurate stock management.

A diagram of a bill

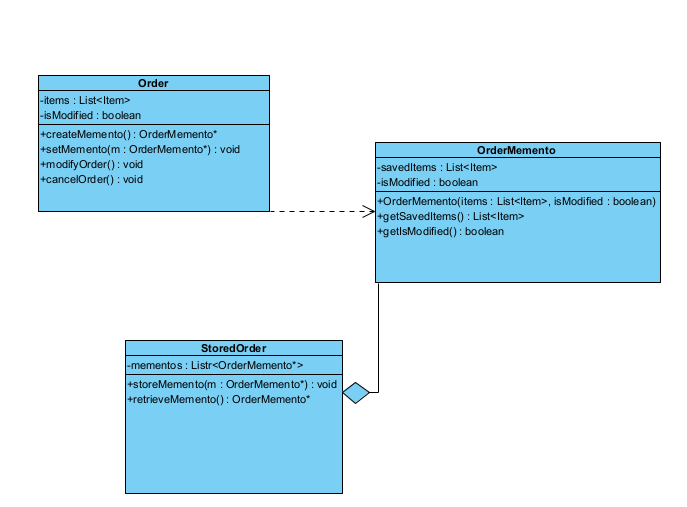
Description automatically generated

9) Composite Pattern - Build up of Bill:  
Problem: Our system needs to represent bills as a hierarchy of components, as well as calculating the final bill.

Solution: We applied the composite pattern to represent bills as a hierarchy of components, allowing the construction and calculation of the final bill by traversing the composite structure.

A diagram of a company

Description automatically generated  
10) State Pattern - Tables seating:  
Problem: Our system needs to manage the state of restaurant tables regarding availability and occupancy.  
  
Solution: We used the state pattern to manage table states, allowing them to transition between "vacant," "occupied," or "reserved" based on customer interactions and reservations.

  
  
11) Memento Pattern - Order preservation and accounting:  
Problem: Our system needs to capture the state of orders, bills, or accounting data at specific points in time.

Solution: We applied the memento pattern to capture the state of orders, bills, or accounting data, enabling order preservation, auditing, and the ability to revert to previous states in case of errors or disputes.

These design patterns helped us to solve our problems in a restaurant management system. They allow our system to be more flexible, it is easier to communicate, overall efficiency has improved and it taught us a lot about designing and implementing systems.

References:

<https://medium.com/@rancelab/10-key-responsibilities-of-a-chef-in-a-restaurant-ed26b4129467>

<https://www.pearsonhighered.com/assets/samplechapter/0/1/3/4/013474506X.pdf>

<https://www.cs.up.ac.za/cs/lmarshall/TDP/TDP.html>

<https://www.bluearrow.co.uk/featured-employers/working-with-sodexo/head-chef-job-description#:~:text=Controlling%20and%20directing%20the%20food,dishes%20ensuring%20variety%20and%20quality>