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** – Decoration of burger/food.
   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** – Generation of plate in the kitchen.
   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** – Different types of chefs
   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** – Communication between kitchen and Waiters
   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 ­**– Customer Handling
   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** – Building up the plate
   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** – Keeping track of inventory
   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** – Build up of Bill
    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** – Tables seating
    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** – Order preservation and accounting
    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.