Cout<<ureFood

Bite by Bite By Bit-by-Bit

Cout<<ureFood

Introduction

Welcome to CoutureFood! Where we bring the dining experience to the digital world

Who are we ?

Couture refers to the design and manufacture of fashionable clothing according to the client's requirements and measurements. Since this is a build-your-own-meal restaurant, we put the couture in food.

Project Overview

This project in particular aims to create a virtual environment that mimics the operations of a real restaurant. Unlike typical tycoon games, this project is designed as a pure simulation, focusing on accurately emulating the functionalities and dynamics of a restaurant's daily operations rather than emphasizing profit-driven or gamified elements.

Core subsystems

Intermediary

Serves as a link between the two subsystems

01

Floor Subsystem

02

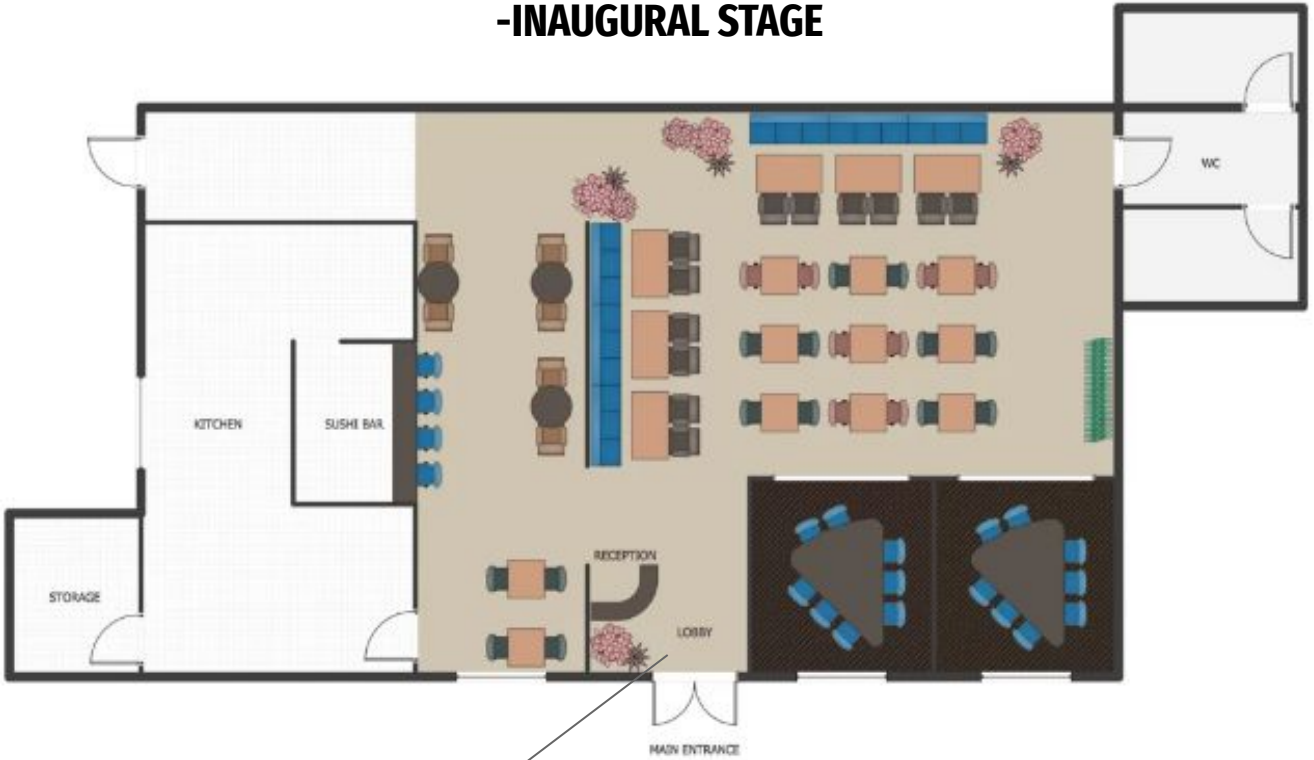
Waiter

03

Kitchen Subsystem

Game Flow

-INAUGURAL STAGE



-MAITRE D

Game Flow
-OPERATIONAL PHASE

Emma

I would like a Burger and chips tonight

70 %

30 %

Josh

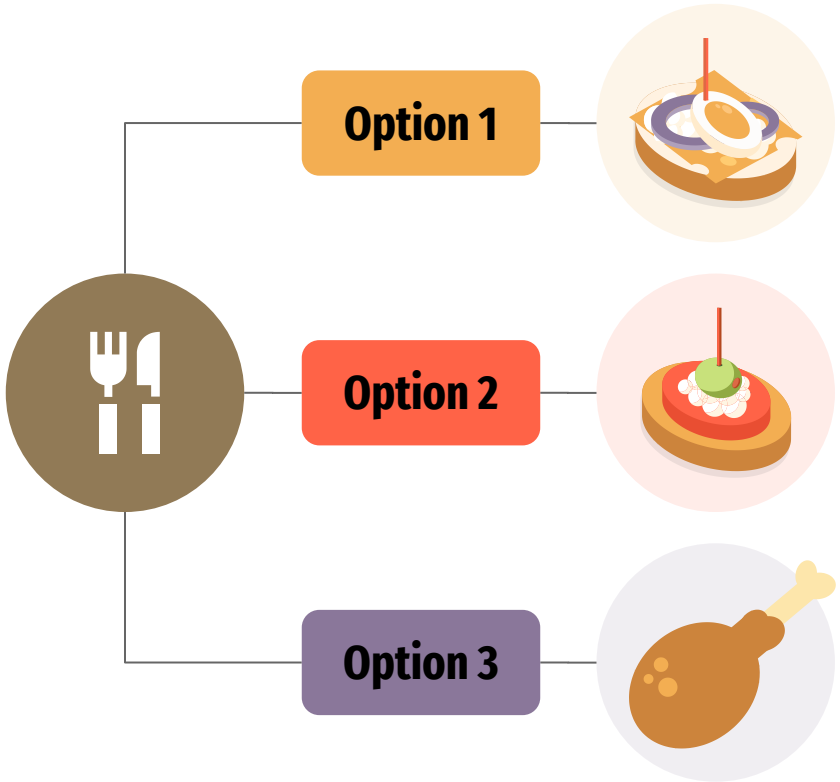
I would like that too!

70 %

30 %

CoutureFood Menu:

"Build your Own" Menu Items:



Chips

Plain or with Sauce
(Jalapeno / Tomato)

Burger

Chicken or Beef

Poultry

Fried or Grilled

Game Flow (cont)

Customer Interaction

Manager makes regular rounds to interact with customers seated at the tables. Engaging in conversations, ensuring their satisfaction, and attending to any specific needs or requests they might have.

50 %

Expectation Management

Managing and setting appropriate expectations for the dining experience.

25 %

75 %

75 %

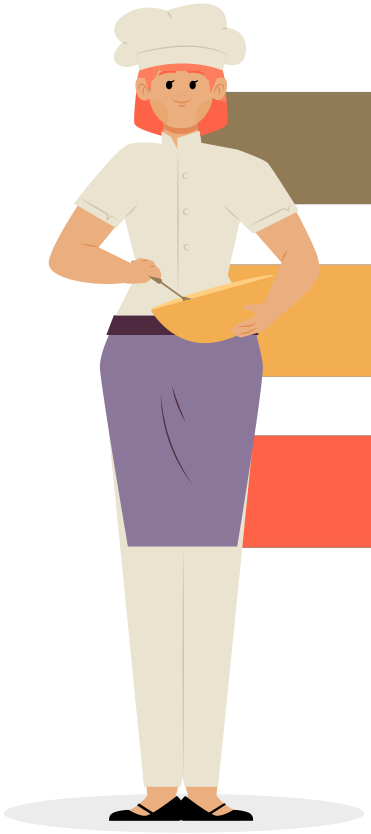
25 %

Complaint Handling

Promptly addressing any complaints or issues that customers might have, taking proactive steps to resolve problems, and ensuring a pleasant dining experience.

50 %

Game Flow (cont)



100 %

Done

50 %

In Progress

75 %

In Progress

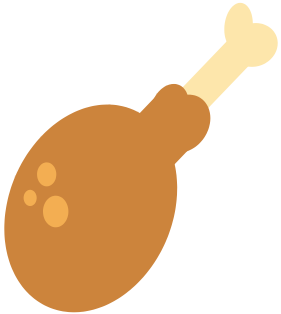
The waiter places an order in the kitchen, it initiates a sequential process within the kitchen subsystem.
As it moves through the stations, different chefs handle and prepare each segment of the meal. The order progresses through state changes at each station, culminating at the Head Chef, responsible for quality assurance of the entire meal.
Once the Head Chef approves the completed order, the waiter is notified to collect it for serving to the customer, signifying the end of the kitchen process. This journey mirrors the systematic preparation and quality check of the meal before it's ready for customer

Game Flow (cont)

Customers		Bill	Amounts Due	
01	Emma		R110	.00
-	Burger & Chips			
02	Josh	R 370,00	R110	.00
-	Burger & Chips			
03	Linda		R150	.00
-	Poultry			

Game Flow (the end)

Food Complaint



Customer Rating

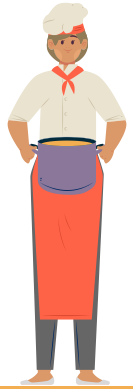
Food was good but a bit burnt.

Service Complaint

Customer Valoration

Waiter was friendly and and professional throughout.

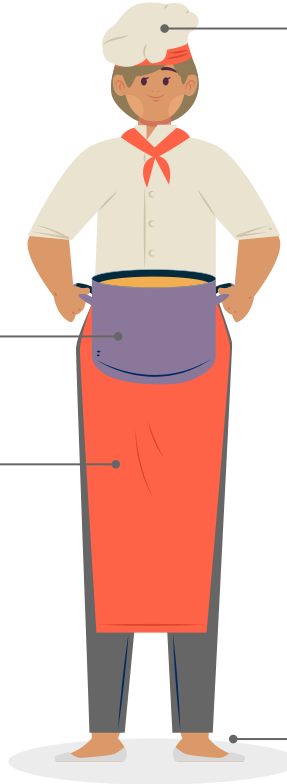
Time Complaint



Customer Valoration

Food took too long to arrive.

Our Ten Platters of C++



01

Singleton

A restaurant has one maitre d' hotel who is responsible for seating customers

02

Chain of responsibility

If a problem cannot be resolved by one staff member, it is passed on to another staff member, ensuring timely resolution

03

State

Meals transition from uncooked to cooked and served, while customers may switch between settling in and ordering

04

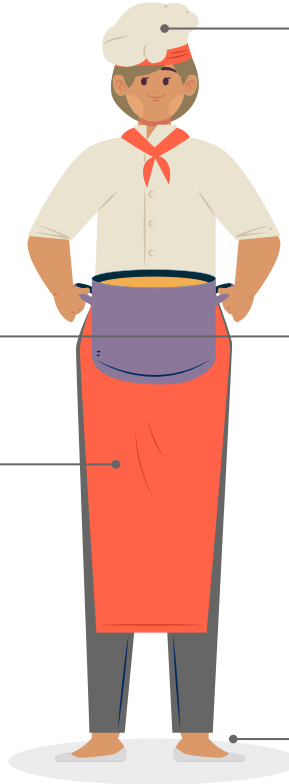
Facade

Users can achieve desired outcomes by following straightforward input instructions

Our Ten Platters of C++ (cont.)

05
Mediator

- Manages communication between staff members.
- Allows addition of new employees without disrupting existing communication channels.



06
Iterator

- Used for traversing tables

07
Composite

- Used for handling customer bills.

08
Strategy

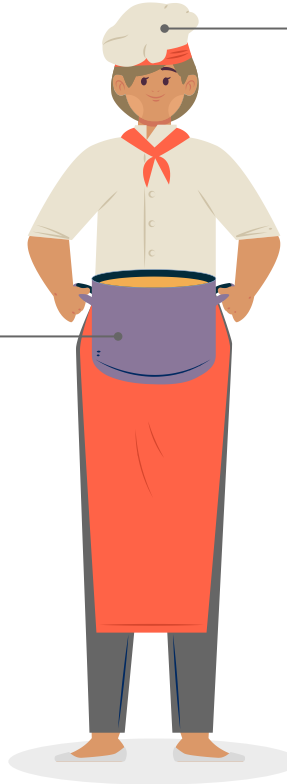
- Used for handling how food items are cooked

Our Ten Platters of C++ (cont)

09

Decorator

AExtends the functionality of food orders by customizing them based on customer preferences.

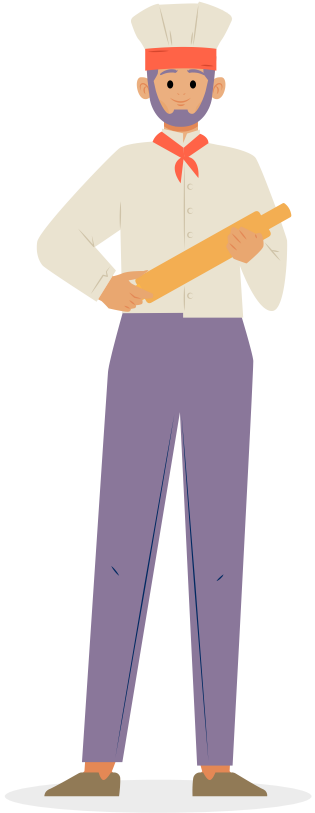


10

Template Method

Allows for the creation of an "algorithm" for preparing meals, with slight variations for different chefs.

Implementation



Okay, Let's serve up some code...