



JNCT

LNCT Group of Colleges

Object Oriented Programming Methodology Case Study

Restaurant Order & Bill Management

TEAM ACHIEVERS

Prof. kalpana M

Team Members 3rd Sem

- | | |
|----------------|--------------|
| 1. KALI KUMARI | 0131CL221045 |
| 2. MD SOHAIB | 0131CL221055 |

INDEX

bdtask
A Leading Software Company



01

NEED OF CASE STUDY

04

ADVANTAGE AND DISADVANTAGE OF CASE STUDY

02

OBJECTIVE OF CASE STUDY

05

PROPOSAL OF CASE STUDY

03

STUDY OF CASE STUDY

06

INPUT-OUTUT OF CASE STUDY

NEED OF CASE STUDY

WHY I NEED TO STUDY THIS CASE STUDY?

1.Learning and Practice:

2. Problem-Solving:

3. Communication Skills:

4Preparation of Interviews:

Why we need to study this case study?

1. Scenario Understanding

2. Understanding User Needs:

3.Practical Application:

4.Creative Thinking:



MOTIVATION BEHIND THIS CASE STUDY

1. Practice Makes Perfect:

- Doing this case study helps you get better at coding and solving problems

2. Real-life Example:

- It's like solving a puzzle based on a real restaurant, so you can see how coding can be useful in everyday situations.

3. Showcasing Your Skills:

- Think of it like creating a scrapbook to show off the cool things you can do with coding.

4. Stay Curious:

- It gives you a reason to explore and learn new things because you want to solve the puzzle.



IS IT USEFUL FOR NATION?

1. *Human Capital Development:*

cultivating a workforce with a strong skill set, contributing to the nation's human capital.

3. *Educational Advancements:*

Well-prepared individuals can contribute to research and development.

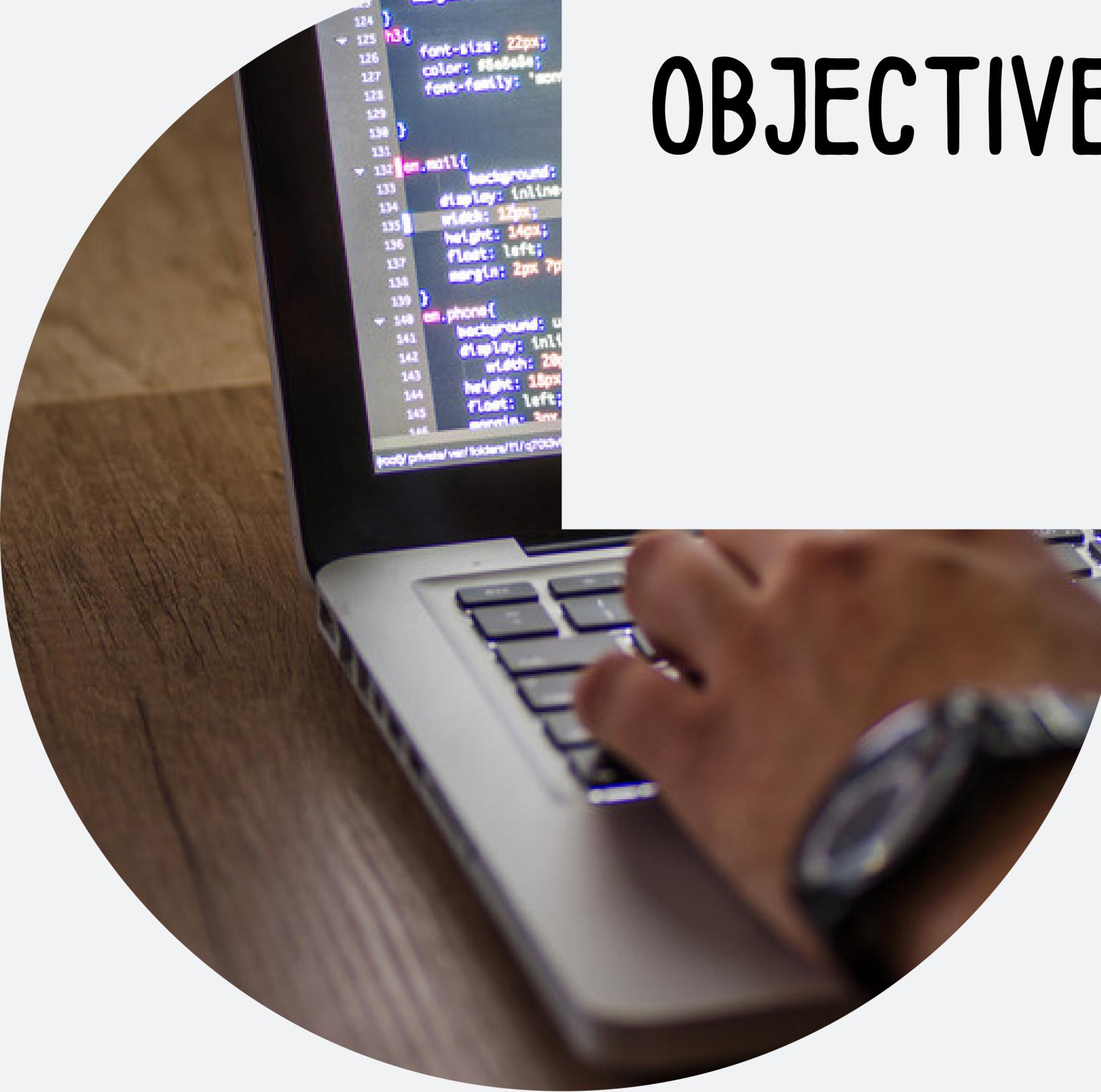
2. *Innovation and Technology:*

contribute to technological advancements within a nation

4 *Global Competitiveness:*

- A skilled workforce can enhance a nation's global competitiveness in the tech industry, attracting investments and creating opportunities for economic growth.

OBJECTIVE OF CASE STUDY



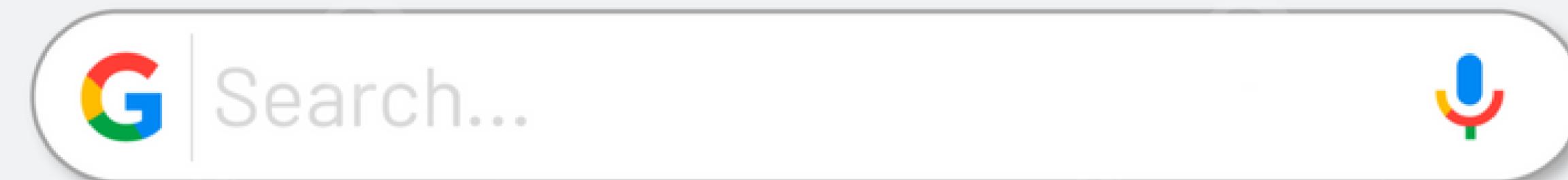
- 1. Efficiency Improvement:** Speeds up the ordering process, reducing wait times for customers and improving table turnover.
- 2. Accuracy Enhancement:** Minimizes errors in orders, leading to fewer mistakes in customer meals and billing.
- 3. Inventory Management:** Helps in tracking inventory levels, enabling better control over stock and reducing wastage.
- 4. Customer Experience:** Enhances the overall dining experience by streamlining the ordering process and ensuring accurate billing.

STUDY OF CASE STUDY

Is case available in Google?

Yes, you can find numerous restaurant management software case studies on Google.

To find these case studies, you can simply search for the software's name in combination with "case study" or "customer success stories" on a search engine like Google.



Additionally, you can also visit the websites of these software providers directly and look for their resources section or customer testimonials to find case studies and success stories associated with their products.

Is any Article present?

**YES THERE ARE RESEARCH PAPER
AVAILABLE**

**Are any research paper available?
YES THERE ARE RESEARCH PAPER
AVAILABLE**

**Websites of
hospitality**

**Industry-Specific
Websites**

Academic Journals

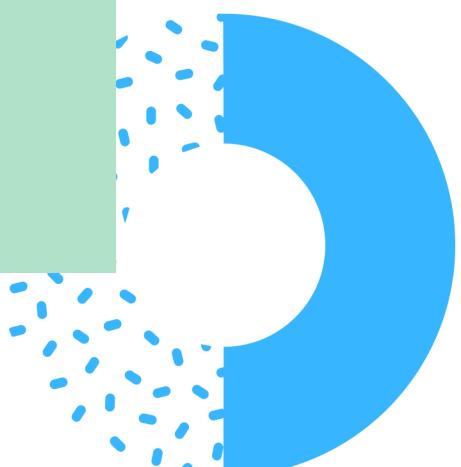
**technology
magazines**

Industry Journals

journal of Foodservice Business
Research
international Journal of Hospitality
Management

Academic Databases

Google Scholar host
Platforms like IEEE Xplore



IS RESTAURANT ORDER AND BILL MANAGEMENT SYSTEM AVAILABE IN THE MARKET?

There are numerous restaurant order and bill management software and apps available in the market.



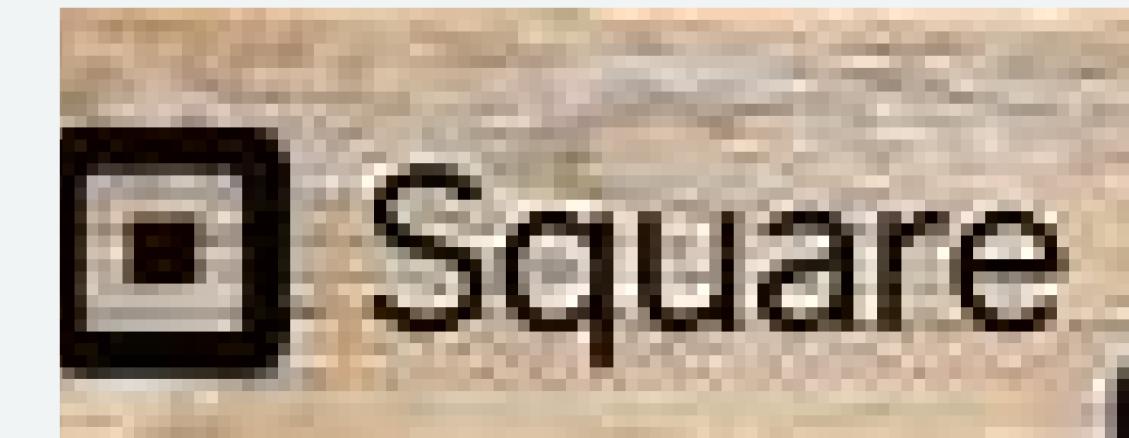
1. Toast

[Toast - Restaurant Management System](#)



2. TouchBistro:

[TouchBistro - Restaurant POS System](#)

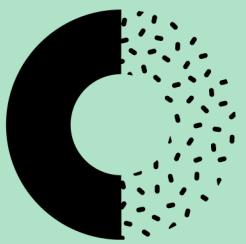


3. Square for Restaurants

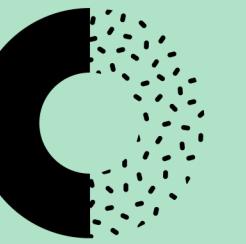
[Square for Restaurants](#)

These solutions help streamline the process of taking orders, managing bills, and improving overall efficiency in a restaurant.

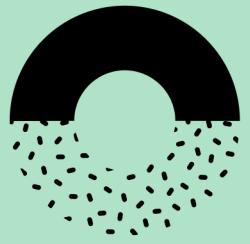
ADVANTAGES



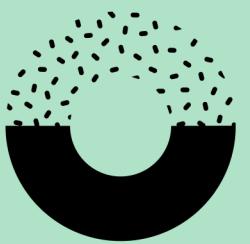
1. Efficiency Improvement:



2. Accuracy Enhancement



3. Inventory Management



4. Customer Experience:

DISADVANTAGES



1. Initial Cost:



2. Technical Issues



3. Learning Curve:



4. Dependency on
Technology:

CLASS AND OBJECT DIAGRAM

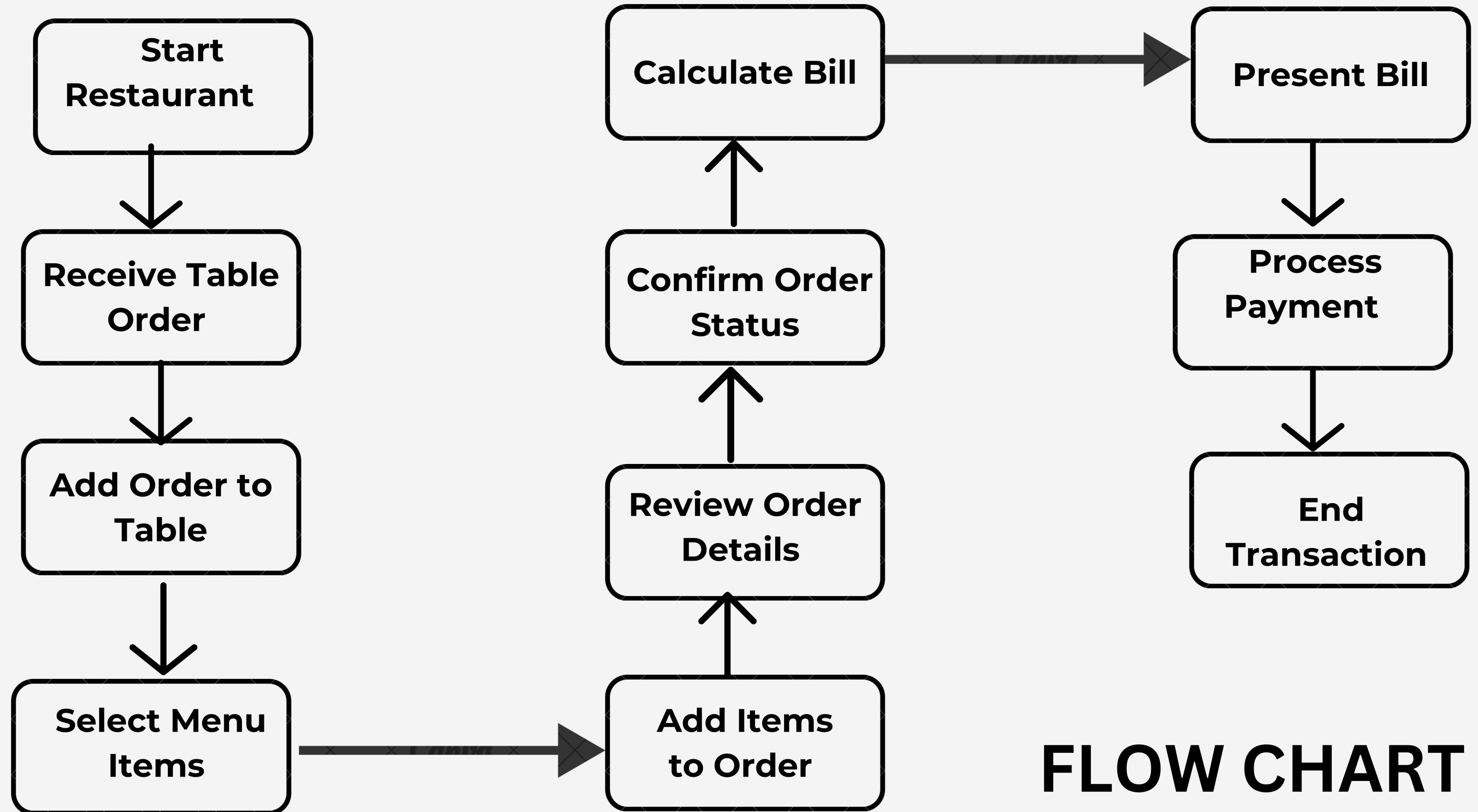
```
Restaurant
-----
name: String
location: String
tables: List<Table>
```

```
MenuItem
-----
itemId: int
name: String
description: String
price: float
updatePrice(newPrice: float): void
```

```
Table
-----
tableNumber: int
status: TableStatus
orders: List<Order>
addOrder(order: Order): void
removeOrder(order: Order): void
calculateTotal(): float
```

```
Order
-----
orderId: int
items: List<MenuItem>
status: OrderStatus
addItem(item: MenuItem): void
removeItem(item: MenuItem): void
```





FLOW CHART

INPUT

```
#include <iostream>
#include <vector>
#include <string>
using namespace std;

class MenuItem {
public:
    string name;
    string description;
    float price;
};

class Order {
public:
    int tableNumber;
    vector<MenuItem> items;
};

class Table {
public:
    int tableNumber;
    bool isOccupied;
    Order currentOrder;
};

class Restaurant {
public:
    vector<Table> tables;
};

void placeOrder(Table& table, const MenuItem& item) {
    table.currentOrder.items.push_back(item);
    cout << "Placed order for: " << item.name << " at table: " <<
        table.tableNumber << endl;
}

void generateBill(Table& table) {
    float totalAmount = 0.0;
    for(const auto& item : table.currentOrder.items) {
        totalAmount += item.price;
    }
}

cout << "Generated bill for table: " << table.tableNumber << ". Total amount: $" <<
    totalAmount << endl;
}
```

```
void processPayment(Table& table, float amountPaid) {
    float totalAmount = 0.0;
    for(const auto& item : table.currentOrder.items) {
        totalAmount += item.price;
    }
    if(amountPaid >= totalAmount) {
        cout << "Payment of $" << amountPaid << " received for table: " <<
            table.tableNumber << ". Payment Successful!" << std::endl;
    } else {
        std::cout << "Insufficient payment for table: " << table.tableNumber << ".
            Payment Failed!" << std::endl;
    }
}

int main() {
    Restaurant myRestaurant;

    // Adding tables to the restaurant
    Table table1;
    table1.tableNumber = 1;
    table1.isOccupied = false;
    myRestaurant.tables.push_back(table1);

    // Simulate order and bill management
    Table& chosenTable = myRestaurant.tables[0]; // Choosing a table for an order

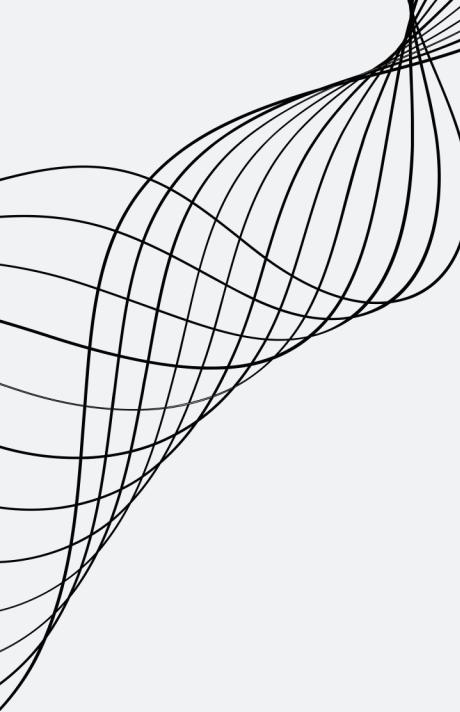
    MenuItem selectedMenuItem;
    selectedMenuItem.name = "Pasta";
    selectedMenuItem.description = "Spaghetti with tomato sauce";
    selectedMenuItem.price = 8.99;

    placeOrder(chosenTable, selectedMenuItem); // Placing an order
    generateBill(chosenTable); // Generating bill for the table
    processPayment(chosenTable, 10.0); // Processing payment

    return 0;
}
```

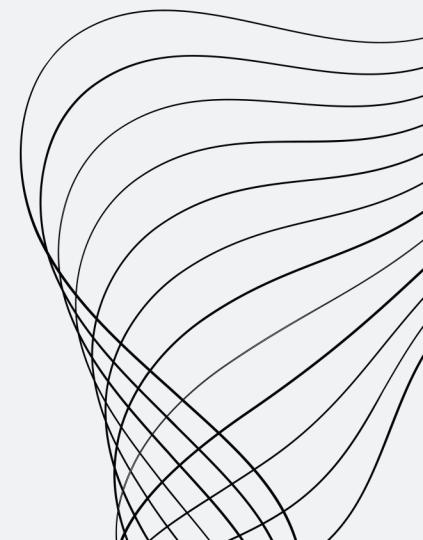
OUTPUT

Placed order for: Pasta at table: 1
Generated bill for table: 1. Total
amount: \$8.99
Payment of \$10 received for table: 1.
Payment Successful!



END OF PRESENTATION

THANK YOU



KALI KUMARI
MD.SOHAIB