

**NANYANG  
TECHNOLOGICAL  
UNIVERSITY**  

---

**SINGAPORE**

## **Project Report**

School of Computer Science & Engineering

CZ2002: Object-Oriented Design & Programming

MACS Group 3

Bian Siyuan U2020849L

Chan Chor Cheng U2040634F

Shrivastava Samruddhi U1923522K

Wang Yujing U2040172J

Zhou Yiqi U2040076J

# Declaration of Original Work

CE/CZ2002 Object-Oriented Design & Programming

Assignment

## APPENDIX B:





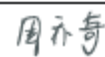
Attached a scanned copy with the report with the filled details and signatures.

### Declaration of Original Work for CE/CZ2002 Assignment

We hereby declare that the attached group assignment has been researched, undertaken, completed and submitted as a collective effort by the group members listed below.

We have honored the principles of academic integrity and have upheld Student Code of Academic Conduct in the completion of this work.

We understand that if plagiarism is found in the assignment, then lower marks or no marks will be awarded for the assessed work. In addition, disciplinary actions may be taken.

Name	Course (CE2002 or CZ2002)	Lab Group	Signature /Date
Bian Siyuan	CZ2002	MACS Group 3	 08/11/21
Chan Chor Cheng	CZ2002	MACS Group 3	 08/11/21
Shrivastava Samruddhi	CZ2002	MACS Group 3	 08/11/21
Wang Yujing	CZ2002	MACS Group 3	 08/11/21
Zhou Yiqi	CZ2002	MACS Group 3	 08/11/21

Important notes:

1. Name must **EXACTLY MATCH** the one printed on your Matriculation Card.

# Table of Contents

<b>Project Report</b>	1
<b>Declaration of Original Work</b>	2
<b>Table of Contents</b>	<b>Error! Bookmark not defined.</b>
<b>Introduction</b>	4
<b>Approach</b>	4
Data Structures	4
<b>OO Design Principles</b>	4
Single Responsibility Principle	5
Open/Closed Principle	5
Liskov Substitution Principle	5
Interface Segregation Principle	5
Dependency Injection Principle	5
<b>OO Concepts</b>	6
Abstraction	6
Encapsulation/Information-Hiding	6
Inheritance	6
Polymorphism	6
<b>Assumptions</b>	6
<b>UML Diagrams</b>	<b>Error! Bookmark not defined.</b>
UML class diagram	7
<b>Test Cases</b>	<b>Error! Bookmark not defined.</b>

## Introduction

This report aims to explain the structural design of the Restaurant Reservation and Point of Sale System (RRPPS). This console-based application performs several functions for restaurant staff, including table and order management, as well as menu modifications and invoice generation.

## Approach

Following the principles of Object-Oriented Design and Programming, this application aims to achieve loose coupling and high cohesion. On a higher level, the program is split into two parts: the back end that handles storage and data processing and the front end which handles user interactions.

## Data Structures

All relevant data including menu, orders, staff information etc. can be saved and loaded with a plain text file, allowing data persistence across multiple sessions.

One of the most important aspects of the restaurant application is the menu. The design of data representation needs to take into consideration the ease-of-use in the back end code during runtime, being able to save and load the data in file format, and the convenience of messaging between the back end and the front end. To allow for seamless messaging between the front end and back end, indexes (IDs) are used for serialization in the front end, while Java object references are used in the back end. An example of this would be the implementation of a HashMap from the ID of each menu item to its corresponding Java object within the Restaurant class which will handle the serialization whereas the back end code uses object references.

Other parts of restaurant data are relatively straightforward. Reservations, orders, table information, and staff information are serialized into a plain text file in self-defined format. Several optimizations are made in regard to the runtime representation of data. *TableManager* uses a TreeSet to store reservations in a time-weighted priority queue, which speeds up query for available tables. Historical orders are only read into memory if the user asks for a sales report, instead of reading a potentially large number of historical orders during program startup. A buffer (inactiveOrders) is maintained during runtime so that the program only needs to write to disk when shutting down.

## OO Design Principles

In this section, the SOLID design principles will be discussed.

## Single Responsibility Principle

The principle is implemented in the design, and is evident in the *Staff* class which is only responsible for the management of staff information. Similarly, the *TableManager* class handles the connection between tables and reservations.

## Open/Closed Principle

In our design, the principle is accomplished by fully utilising the abstract class *AbstractMenuItem* as the base class which is extended to implement *Category*, *MenuBundle*, and *MenuLeaf* classes. Inside the base class, the `print()` method is intended to print the name and description for each item. Instead of modifying this method, it is overridden in the subclasses to extend the printing functionalities with different formats.

## Liskov Substitution Principle

This idea embodies the design of the *Category* class which contains an array of *AbstractMenuItems*. Each *AbstractMenuItem* within the array is substitutable by any of its derived classes i.e. *MenuLeaf* or *MenuBundle*. When an *Order* object iterates through all *MenuItem* in an order, it is able to call `getTotalPrice()` without caring about whether the item is a *MenuLeaf* or *MenuBundle*.

## Interface Segregation Principle

This principle is applied in the design of *MenuItem* and *MenuList*. *MenuItem* contains methods about name, description, and most importantly price, while *MenuList* contains methods about managing children items. Different types of menu related classes can decide which interface to implement. For example, *Category* (which does not have a price) only needs to implement *MenuList*, while *MenuBundle* implements both *MenuItem* and *MenuList*.

## Dependency Injection Principle

In our design, as mentioned earlier, *AbstractMenuItem* is an abstract class. This abstract class facilitates the communications between the front end and different concrete menu related classes such as *MenuBundle* and *MenuItem*.

# OO Concepts

## Abstraction

This concept aims to reduce the complexity of the code. It refers to the creation of conceptual boundaries that distinguish essential characteristics from the perspective of the viewer. The *AbstractMenuItem* base class is able to do this as an abstract class.

## Encapsulation/Information-Hiding

An example of encapsulation being applied in our design is the *SalesReport* class, which only exposes a method for adding a historical order to the report and printing the report. This hides the internal representation and maintains the invariance that only historical orders in a certain time period can be added.

## Inheritance

The use of this OO concept prevents extensive duplication throughout the code, by reusing a parent class' code, and also inheriting the properties and methods of the parent class. *AbstractMenuItem* is the parent class of *MenuBundle* and *MenuLeaf*. This provides a unified interface and enables code reuse.

## Polymorphism

An example of polymorphism in our code is when the front end calls the `print()` method on an *AbstractMenuItem* object, it will resolve to the specialized `print()` method in different subclasses like *MenuBundle* and *MenuLeaf*.

## Assumptions

There are several assumptions made in this design, of which some are listed below:

- All the restaurant staff have the same level of authority and access to all functions equally. Thus, there is no access control or authentication to use the application.
- Membership is a boolean. There are no 'levels' of holding membership in consideration.
- An order is considered 'inactive' once the invoice is printed, which in real life, could precede the table being vacated.

- In the back end, the assumption is that every order has a reservation. While the user does not need to create a reservation for walk-in customers, a ‘ghost’ reservation is created in the back end alongside the order for walk-in customers.
- In the front end, the assumption is that promotional bundles can only exist in a dedicated category called “Combo” and the restaurant menu only contains menu categories (does not directly contain individual items). Although the backend actually supports placing bundle items anywhere and placing single items directly under the restaurant menu, the assumption is made to simplify the front end code.
- The front end does not provide an interface for staff or table information to be added or changed. If there is such a need, the plain text file storing staff information can be directly edited.

## UML Diagrams

### UML class diagram

Figure 1.1 focuses on showing the dependencies between classes, while Figure 1.2 shows the full UML class diagram of the application including all attributes, methods, class stereotypes, multiplicities and connections while excluding dependencies.

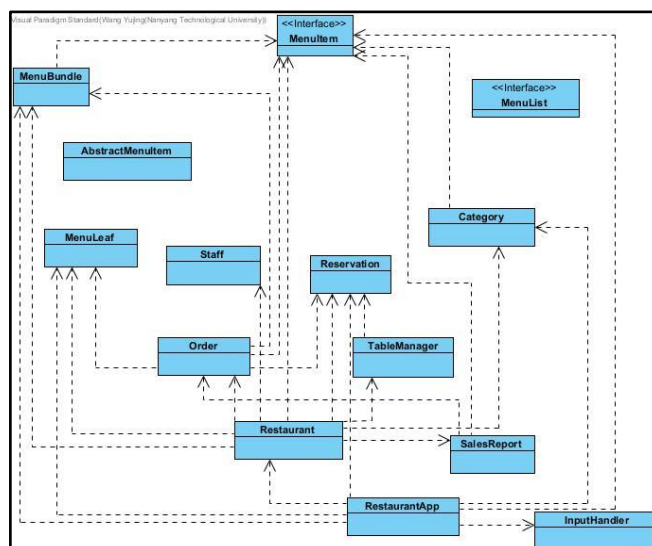


Figure 1.1 Simplified UML class diagram

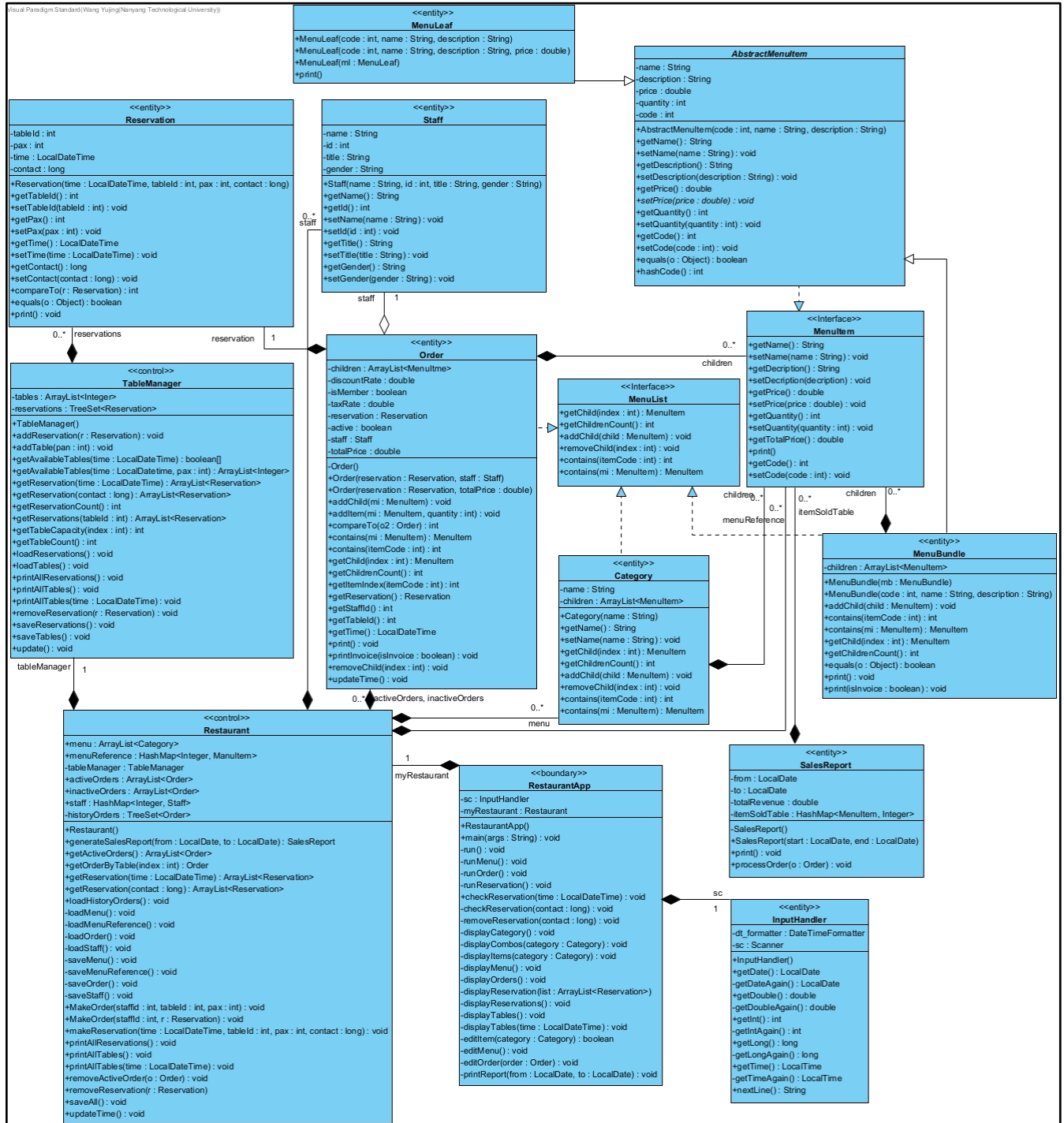
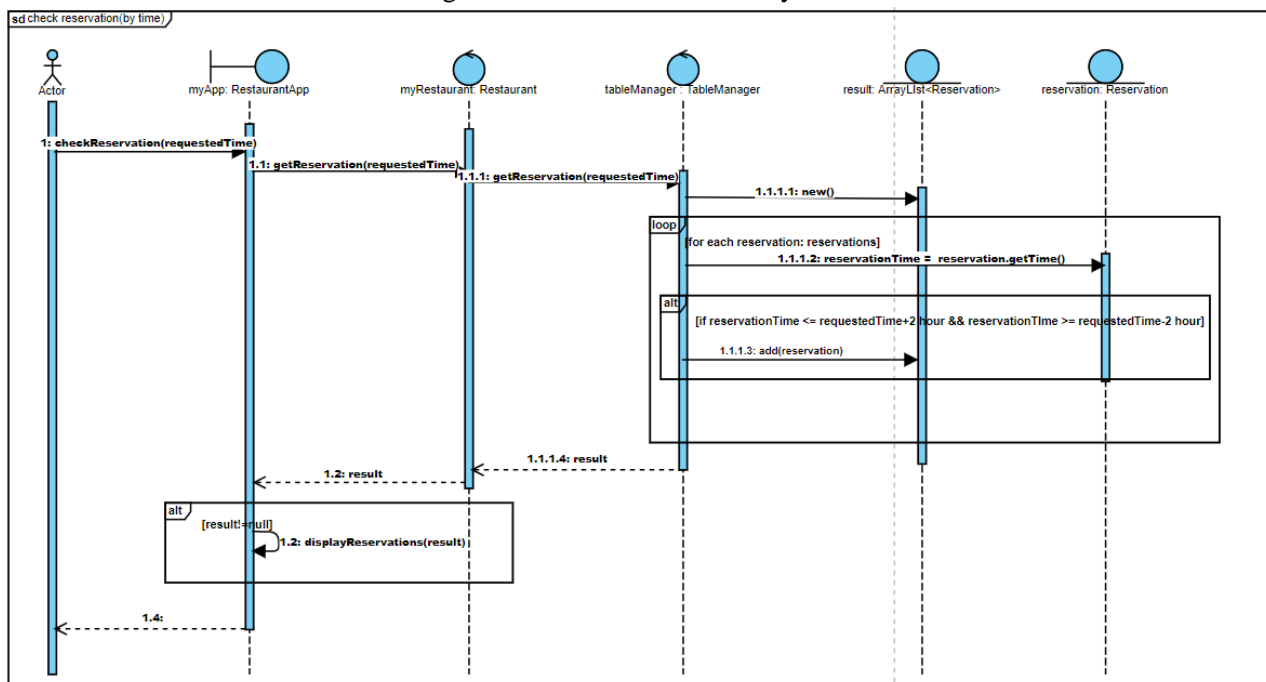
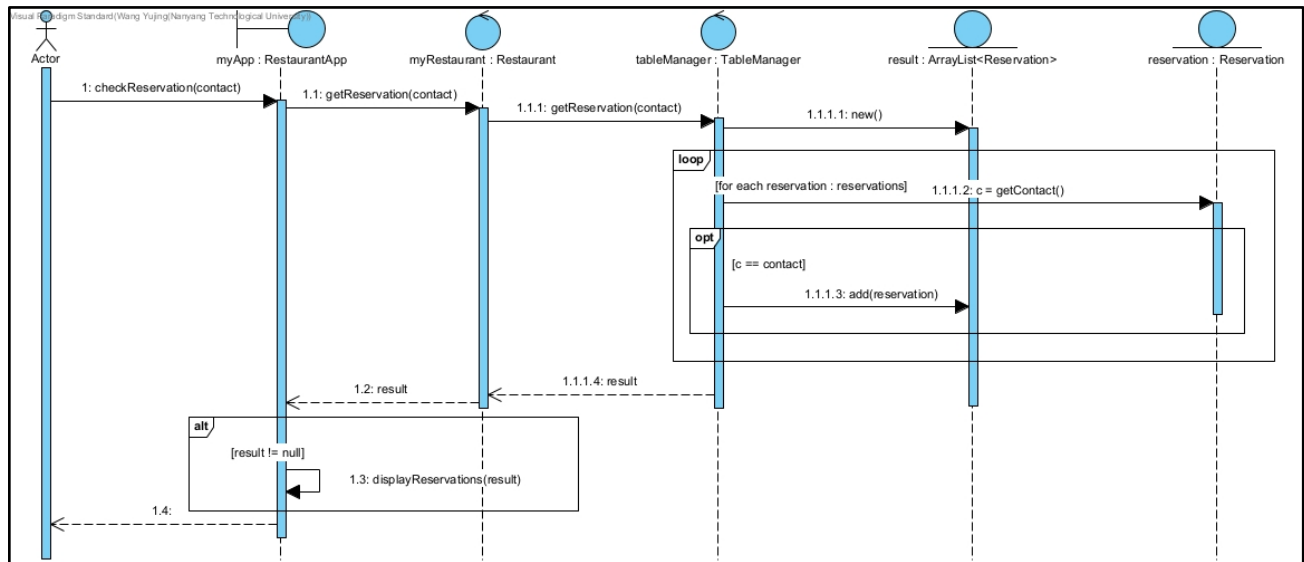


Figure 1.2 Detailed UML class diagram



## UML Sequence Diagram

Figure 2.1 and 2.2 show the sequence diagrams for checking reservation by contact numbers and time respectively. Figure 2.3 and 2.4 show how reservations are removed by contact number and time expiry respectively. For expired reservations to be removed, an update() method defined within TableManager will be called by other functions to refresh the reservations status before any CRUD operations to be done on reservations.



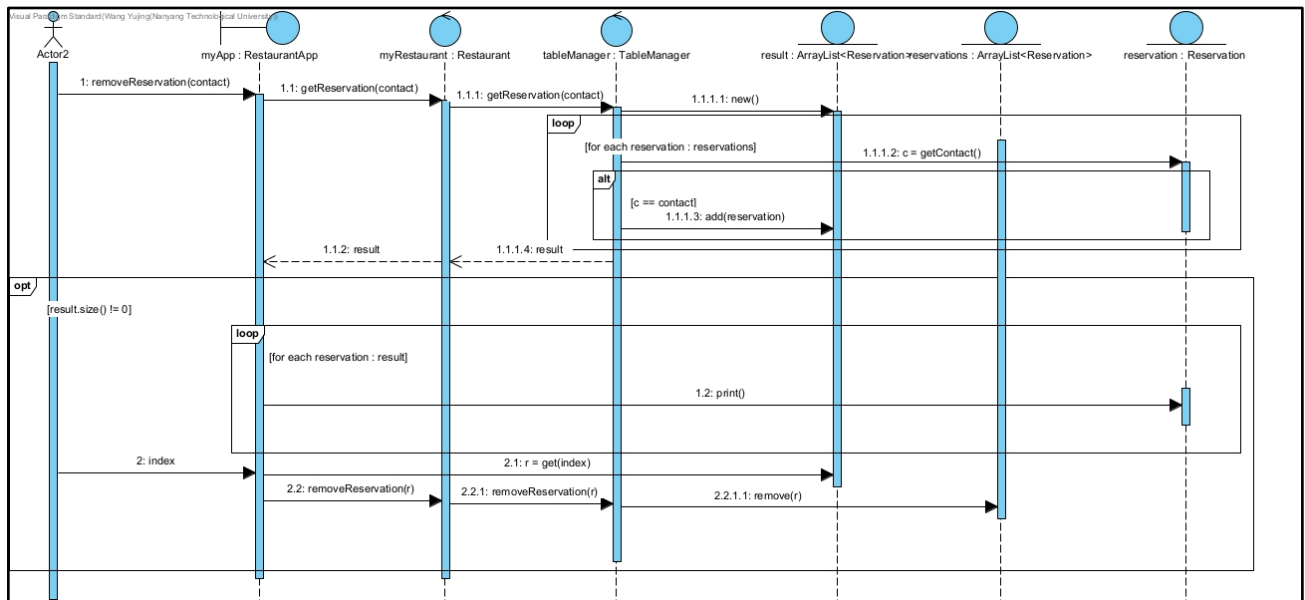


Figure 2.3 Remove reservation by contact

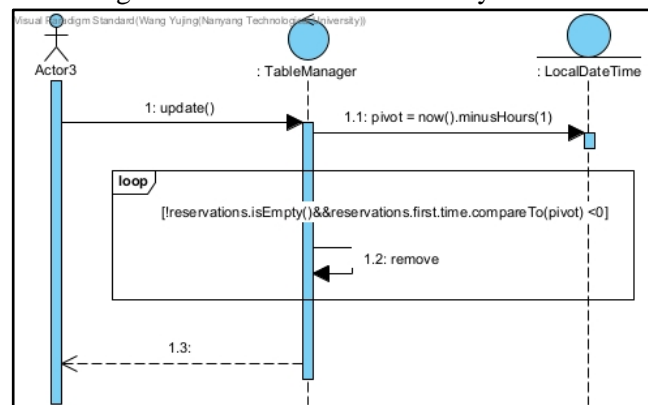


Figure 2.4 Update function to remove expired reservation(s)

## Test Cases

Link to the demonstration video : <https://www.youtube.com/watch?v=EYSSMomhDoQ>

Test cases covered in the video in order of appearance

- View restaurant menu
- Add menu item
- Add promotional item
- Remove menu item
- Remove promotion item
- Check reservation by contact number
- Remove reservation
- Create reservation (restaurant is full)
- Create reservation (restaurant has empty tables)
- Check reservation by time
- Create order
- Add items into order
- Remove item from order

- View order
- Check table availability (now)
- Print order invoice (including asking for membership status)
- Check table availability (table vacated after printing invoice)
- Check reservations by time (expired reservations are removed)
- Print sales report

Test case not demonstrated in video: updating menu and promotional items (shown below)

```
Menu
Appetiser
1324 Truffle Mushroom Soup 6.00
      Button Mushroom, Truffle Oil, Fresh Cream and Croutons
1895 Roasted Tomato Soup 5.00
      Oven-roasted Tomatoes, Garlic and Croutons
3442 Charcoal-grilled Surme Ika (whole) 15.00
      Chargrilled Japanese Flying Squid in Teriyaki Sauce
Salad
0241 Smoked Duck Salad 9.00
      Smoked Duck Breast, Japanese Cucumber, Cherry Tomato
7853 Classic Casesar Salad 8.00
      Baby Romaine Lettuce, Egg, Croutons, Bacon and Parmesan Cheese
Pasta
0082 Carbonara 15.00
      Spaghetti, Bacon, Mushroom and Ham in Cream Sauce
4288 Truffle Cheese Ravioli 12.00
      Truffle Cheese, Parmesan Cheese, Truffle Mushroom Ravioli
Meat
0951 Signature Roasted BBQ Pork Ribs 19.00
      Coleslaw, Cajun Fries, Corn Cob, Mesclun Salad with BBQ Sauce
0123 Argentine Black Angus Ribeye Prime Steak (200g) 24.00
      Seasonal Vegetables, Potato Au Gratin, Pesto Cream and Black Pepper Sauces
Dessert
0190 Tiramisu 8.00
      Classic Italian Dessert made with Espresso-soaked Ladyfiners and Mascarpone Cream
0402 English Carrot Cake 9.00
      with Vanilla Ice Cream
Drinks
0627 Milky Popcorn 6.00
      Japanese Sencha with Roasted Rice, popcorn with Butterscotch Aroma
0333 Kiss On the Lips 6.00
      Grenadine, Soda, Orange Juice
Combo
0849 BBQ Pork Ribs Combo 22.00
      Truffle Mushroom Soup
      Signature Roasted BBQ Pork Ribs
0748 Ravioli Combo 16.00
      Classic Casesar Salad
      Truffle Cheese Ravioli
```

View original menu

```
Choose Category:4
0. Go back
1. Add item
2. Update item
3. Remove item
4. Remove this category
Choose option:2
0951 Signature Roasted BBQ Pork Ribs 19.00
      Coleslaw, Cajun Fries, Corn Cob, Mesclun Salad with BBQ Sauce
0123 Argentine Black Angus Ribeye Prime Steak (200g) 24.00
      Seasonal Vegetables, Potato Au Gratin, Pesto Cream and Black Pepper Sauces
Enter item code:
123
0. Go back
1. update name
2. update description
3. update price
Choose option: 3
Enter new price: 25
```

Update menu item

```
Choose Category:7
0. Go back
1. Add item
2. Update item
3. Remove item
4. Remove this category
Choose option:2
0849 BBQ Pork Ribs Combo 22.00
      Truffle Mushroom Soup
      Signature Roasted BBQ Pork Ribs
0748 Ravioli Combo 16.00
      Classic Casesar Salad
      Truffle Cheese Ravioli
Enter item code:
849
0. Go back
1. update name
2. update description
3. update price
4. add item
5. remove item
Choose option: 4
Enter item code: 190
```

Update promotional item

```
Menu
Appetiser
1324 Truffle Mushroom Soup 6.00
      Button Mushroom, Truffle Oil, Fresh Cream and Croutons
1895 Roasted Tomato Soup 5.00
      Oven-roasted Tomatoes, Garlic and Croutons
3442 Charcoal-grilled Surme Ika (whole) 15.00
      Chargrilled Japanese Flying Squid in Teriyaki Sauce
Salad
0241 Smoked Duck Salad 9.00
      Smoked Duck Breast, Japanese Cucumber, Cherry Tomato
7853 Classic Casesar Salad 8.00
      Baby Romaine Lettuce, Egg, Croutons, Bacon and Parmesan Cheese
Pasta
0082 Carbonara 15.00
      Spaghetti, Bacon, Mushroom and Ham in Cream Sauce
4288 Truffle Cheese Ravioli 12.00
      Truffle Cheese, Parmesan Cheese, Truffle Mushroom Ravioli
Meat
0951 Signature Roasted BBQ Pork Ribs 19.00
      Coleslaw, Cajun Fries, Corn Cob, Mesclun Salad with BBQ Sauce
0123 Argentine Black Angus Ribeye Prime Steak (200g) 25.00
      Seasonal Vegetables, Potato Au Gratin, Pesto Cream and Black Pepper Sauces
Dessert
0190 Tiramisu 8.00
      Classic Italian Dessert made with Espresso-soaked Ladyfiners and Mascarpone Cream
0402 English Carrot Cake 9.00
      with Vanilla Ice Cream
Drinks
0627 Milky Popcorn 6.00
      Japanese Sencha with Roasted Rice, popcorn with Butterscotch Aroma
0333 Kiss On the Lips 6.00
      Grenadine, Soda, Orange Juice
Combo
0849 BBQ Pork Ribs Combo 22.00
      Truffle Mushroom Soup
      Signature Roasted BBQ Pork Ribs
      Tiramisu
0748 Ravioli Combo 16.00
      Classic Casesar Salad
      Truffle Cheese Ravioli
```

View updated menu