Contents

[Introduction 1](#_Toc534473828)

[Use Cases summarizing the requirements of the program 1](#_Toc534473829)

[A Use Case Diagram for processing an order 2](#_Toc534473830)

[A Class Diagram of the intended system 3](#_Toc534473831)

[A Sequence Diagram for processing a customer’s order 3](#_Toc534473832)

[Lesson learned 4](#_Toc534473833)

# Introduction

A small console application has been developed using C# object oriented programming language in Visual Studio IDE. The application simulates the takeaway shop where customer orders food item depending upon the availability of some packages offer in the shop.

# Use Cases summarizing the requirements of the program

It is a small console program for Sab required to handle the customer order for pizzas and pasta. The customer choose the listed packages or item of food from the menu. The customer as actor interacts with the system. The application will calculate the amount based on the order count and price of each item or packages. The payment information displays the total amount of payment received for pizza order and pasta order with total count of pizzas and pasta sold. The program should be flexible enough to include additional items and packages at later period. Once the order is processed, bill is issued to customer and the program returns to the main menu. For that, an interface has to be used among with polymorphism for the various food item classes. This will create minimum changes in code for adding in future period of time.

The customer is able to view the packages available that will provide flexible option to choose the item and finally make the payment for order to be deliver to customer. The customer can be the bill with the total item order with each item price and total amount with the offer if received based on the packages selection. Depending upon the food packages selection either pasta or pizza or both, combination of complementary soft drinks or dessert item or the garlic breads are available.

Each packages or food item is provided with the unique code number to select from the menu. For example:

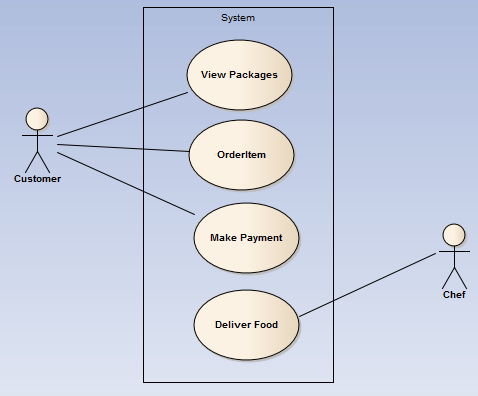
0: For exit the program

1: For pizza selection

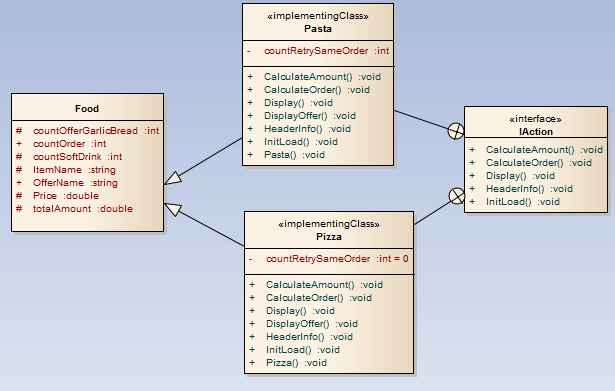
2: For pasta selection

3: For display order summary and clear all the item selected i.e. make object null.

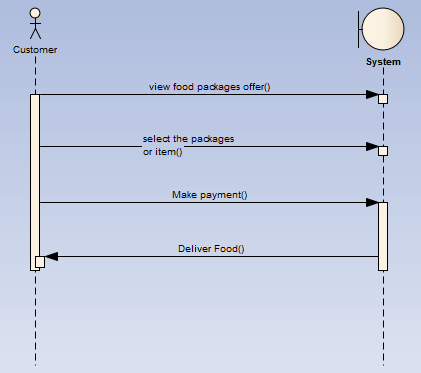
# A Use Case Diagram for processing an order



# A Class Diagram of the intended system



# A Sequence Diagram for processing a customer’s order



# Lesson learned

In order to develop this console application we need to understand the client requirements, design concepts and various aspects related to program development. The most important are the design techniques and analysis of user stories and understanding of object oriented concept of programming that includes polymorphism, inheritance has been really helpful to implement in this program. Similarly, use of interface is also required. After reading user stories, we have to consider designing of class diagrams, strategy of program structure design, class, methods, and properties and so on. Conversion of exact user stories and requirements into the program model structure is crucial part. An Enterprise Architect has been used for constructing the sequence diagrams, use case diagrams and class diagrams. It is necessary to determine how the customer will interact with the system.

This simple application teaches the need of software engineering analysis and design. Before diving into the coding part, analysis and design concept, recognition of problem and come up with the refine solutions plays a crucial part. Knowledge of an architecture design, actors involved, sequence of information flow among the entities, logical structure, functionality are basic lesson to be gain. Every program contains error or bugs. It is necessary to do testing and debugging of the system before ready for use of user. For that, we have to determine possible cases of user input and simple design for easy interaction to the system. A documentation has to develop for the user to understand and its ways of using the system.