

**UNIVERSITI TEKNOLOGI MALAYSIA, JOHOR BAHRU**

**FACULTY OF COMPUTING**

**DATA STRUCTURE AND ALGORITHM**

**(SCSJ2013 - 08)**

|  |
| --- |
| Mini Project Documentation  “Master Burger Ordering Service” |

**SEMESTER I - 2015/2016**

**By:**

MUHAMMAD ZULKIFLI BIN ZAINUL (A14CS0173)

Contents

[PART 1: INTRODUCTION 1](#_Toc439197057)

1.1 [Project Synopsis 1](#_Toc439197058)

1.2 [Project Objective 2](#_Toc439197059)

[PART 2: SYSTEM ANALYSIS AND DESIGN 3](#_Toc439197060)

[2.1 Class Diagram 3](#_Toc439197061)

[PART 3: SYSTEM PROTOTYPE 4](#_Toc439197062)

# PART 1: INTRODUCTION

## 1.1 Project Synopsis

Nowadays, almost everything we do in our daily life is computerized including what we choose to eat. So, my partner and I have decided to make a fast food ordering system. Our system is called “Master Burger Ordering Service”.

Our system includes the use of what we learn from the subject Data Structure and Algorithm which is Queue. For this system, Queue is implemented at the ordering process. For example, customer ordered three types of food, the menu that the customer have chosen is stored inside a queue which follows the concept first-in, first-out (FIFO). The type of queue that we use is pointer-based which is linear linked list.

We decided to use the pointer-based queue because it is more straightforward compared to array-based. It have two external pointers **front** and **back** that are used to point to the first node in the queue and point to the last node in the queue.

These are some of the operation that we applied in the system:

* createQueue() : Creates an empty Abstract Data Type (ADT) of the Queue type which will store what the customer orders
* isEmpty() : Determines whether the is an order in the ADT
* enQueue() : Inserts a new item (customer order) in the back position
* deQueue() : Deletes an item (customer order) from the front position

We also implement a sorting algorithm in our system, which we use Bubble Sort. There are two types of sorting that we use which is to sort the menu/order by names and by price. The sorting by name will arrange the menu/order alphabetically while the sorting by price will sort the price from low to high.

## 1.2 Project Objective

These are the objectives of the project:

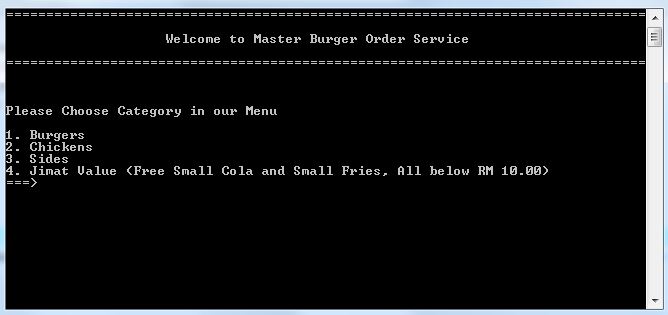
* To ensure that whenever there are new menu being ordered by the customer, the system will store it in an order that is easy to organize and will be arranged neatly.
* To make sure that the system can sort the type of menu by their name and price accordingly so that it is organized.
* To allow customer to cancel their order if they want to do so.
* Customer can also delete or add order even though the receipt is already being displayed.

# PART 2: SYSTEM ANALYSIS AND DESIGN

## 2.1 Class Diagram

|  |
| --- |
| Queue |
| nodeQ \*backPtr  nodeQ \*frontPtr  nodeQ \*newPtr  total : float  num : int |
| Queue()  ~Queue()  isEmpty() : bool  createQueue() : void  destroyQueue() : void  enQueue(string, float) : void  deQueue() : void  displayList() : void  getFront() : string  getRear() : string  bubbleSort(string, float, int) : void  pribubbleSort(string, float, int) : void  chooseMenu(string, float, int) : int |

# PART 3: SYSTEM PROTOTYPE



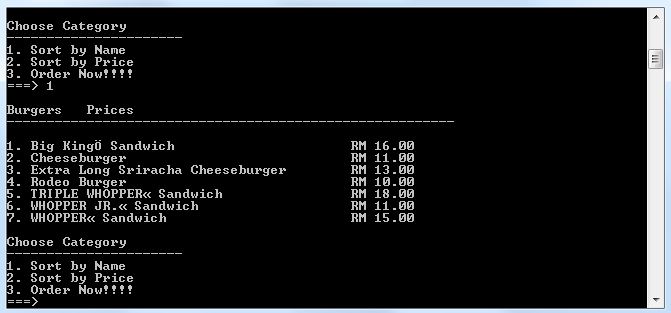
**Screen 1:** Main menu

**Screen 1:** The user is given a choice of menu category and must insert an integer value in the range 1-4 to continue.



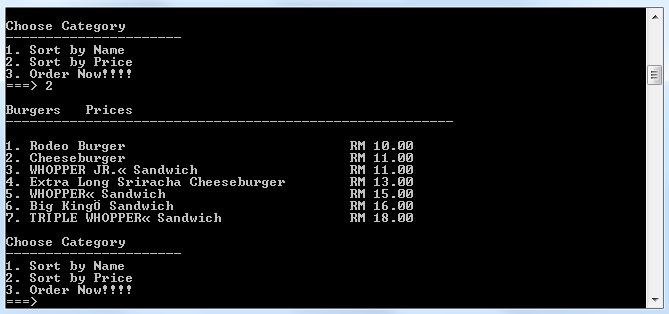
**Screen 2:** Ordering fast food

**Screen 2:** This is the example of output when user chooses 1 which is Burgers. Then, the user is given 3 choice which is to sort the menu by name or by price and also for ordering the menu.



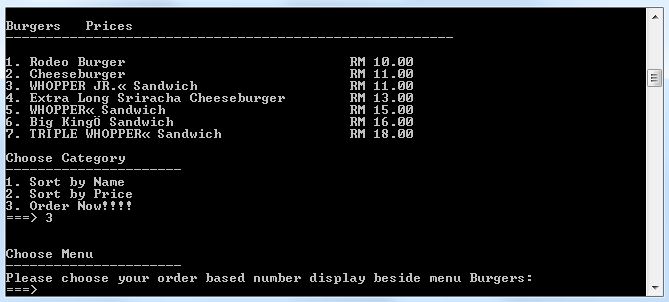
**Screen 3:** Sort by Name

**Screen 3:** The screen shows the result when user chooses 1, which is to sort the menu by name. We can see that the menu list was rearrange alphabetically.



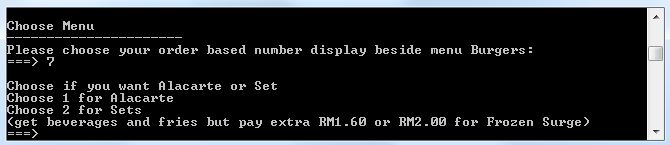
**Screen 4:** Sort by Price

**Screen 4:** Now, the screen shows the output when user chooses 2, which is to sort the menu by price. The menu list is now arranged by the value of price for each item. It is sort from lower price to higher price.



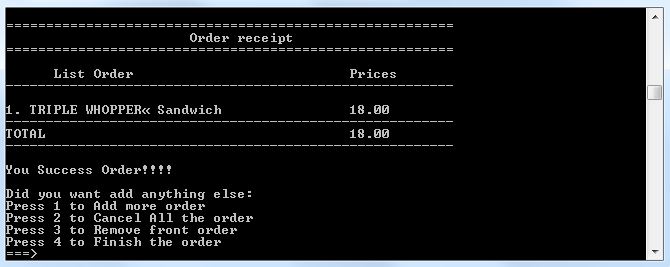
**Screen 5:** Order Now

**Screen 5:** This is the output when user chooses 3, which is to order the food.



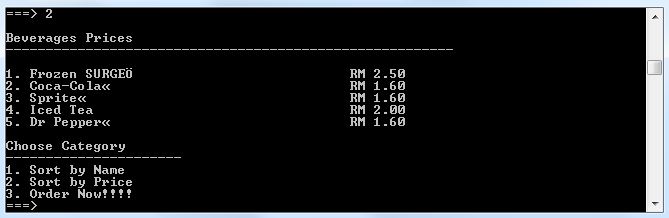
**Screen 6:** Ala Carte or Set

**Screen 6:** For example, when user chooses 7 from the menu which is to order “TRIPLE WHOPPER Sandwich”, the system will ask the user to choose between Ala Carte or Set.



**Screen 7:** Receipt

**Screen 7:** This is the output when user chooses 1, which is Ala Carte. Then, there will be 4 option for the user to choose that is 1 to add more order, 2 to cancel all the order, 3 to remove front order and 4 to finish the order.



**Screen 8:** Set

**Screen 8:** from screen 6, the user chooses 2 which is Set. The system will display list of beverages for user to choose.



**Screen 9:** Receipt

**Screen 9:** This is the output of the receipt when user to 3 to order and choose 3 again for a beverage named “Sprite” from **Screen 8**.