



**UITM PERAK BRANCH TAPAH CAMPUS**

**COLLEGE OF COMPUTING INFORMATIC AND MATHEMATIC**

**PROPOSAL OF DR. AZHAR HEALTHCARE PHARMACY**

**GROUP PROJECT CSC248**

<b>GROUP MEMBER</b>	<b>STUDENT ID</b>
Muhammad Akmal Fikri Bin Asma Zaki	2023466942
Muhammad Amsyar Bin Jamaluddin	2023848218
Muhammad Qais Qaiman Bin Mohd Nafiah	2023696268
Wan Muhammad Hanif Bin Wan Muhammad Jaafar	2023863368

**GROUP : CDCS1103A**

**PREPARED FOR : DR.NURSYAHIDAH BINTI ALIAS**

## **MEDICAL INDUSTRY**

### **Table of Content**

1.0 Introduction .....	3
2.0 Distibution of work among team members .....	5
3.0 Complete Coding according by Classes .....	6
3.1 Class Medicine .....	6
3.2 Class Linked List .....	10
3.3 Class Queue .....	16
3.4 Class Node .....	16
3.5 Class MedicineMainLinkedList .....	16
3.6 Class MedicineMainQueue .....	25
4.0 Sample Output .....	33
4.1 MedicineMainLinkedList .....	33
4.2 MedicineMainQueue .....	42
5.0 Conclusion .....	52

## 1.0 Introduction

Efficient medicine inventory and transaction data management is crucial for smooth movement of medicine stock and providing excellent customer service in healthcare industry which has become quite fast-paced. Dr. Azhar Healthcare Pharmacy suffers from difficulty keeping accurate records of available medicines, recording sales transactions, monitoring stock levels for timely restocking of medicines. In order to combat these problems, a custom application will be programmed in Java.

This is an application which will provide all the medicine-related information i.e medicine type, dosage, instructions, price and quantity in stock as a whole on a single platform. Moreover, it will also make the process of tracking sales, updating inventory levels, as well as identifying items with low stock for restocking process easier. The application will at the end of each month generate comprehensive reports for the pharmacy to help it spot its sales date and make informed decisions on inventory management.

Using this application, Dr. Azhar Healthcare Pharmacy will gain better service quality and customer satisfaction along with better operational efficiency and stock visibility and availability of essential medicines for customers.

This application will be able to:

1. Store the medicine information in the application.
2. Search medicine quantity below 10.
3. Calculate the total price of medicine for specific medicine.
4. Calculate the total price of all medicine inventory.
5. Split the medicine data according medicine type.
6. Search and update the medicine quantity based on user input and store at UpdatedData.txt

**OBJECT:** Medicine

**ATTRIBUTES:** Medicine Name, Medicine Type (Fever / Cough / Flu / Allergy / Vitamin), Medicine Dosage, Medicine Quantity, Medicine Instruction, and Medicine Price

**METHODS:** Accessor method, mutator method, processor method and toString method

List of group members:

Muhammad Akmal Fikri Bin Asma Zaki	2023466942
Muhammad Amsyar Bin Jamaluddin	2023848218
Muhammad Qais Qaiman Bin Mohd Nafiah	2023696268
Wan Muhammad Hanif Bin Wan Muhammad Jaafar	2023863368

**List of processing:**

1. Store the medicine information in the file.
2. Search medicine quantity below 10.
3. Calculate the total price medicine for specific medicine
4. Calculate the total price of all medicine inventory
5. Split the medicine data according medicine type
6. Search and update the medicine quantity base on user input and store at UpdatedData.txt

**Data in input file**

Paracetamol;Fever;500;10;Take one tablet every 4-6 hours as needed;2.50  
 Cetirizine;Allergy;10;15;Take one tablet daily;5.00  
 Vitamin C;Vitamin;500;20;Take one tablet after meals daily;7.00  
 Cough Syrup;Cough;10;100;Take 10 mL three times a day;4.50  
 Ibuprofen;Fever;200;12;Take one tablet every 6-8 hours as needed;3.00  
 Amoxicillin;Infection;500;14;Take one capsule every 8 hours for 7 days;10.00  
 Loratadine;Allergy;10;10;Take one tablet daily;4.00  
 Azithromycin;Infection;250;6;Take two tablets on day 1 then one tablet daily for 4 days;12.00  
 Multivitamin;Vitamin;1;30;Take one tablet daily with meals;8.50  
 Dextromethorphan;Cough;15;120;Take 15 mL every 6 hours as needed;6.00  
 Losartan;Blood Pressure;50;30;Take one tablet daily;15.00  
 Metformin;Diabetes;500;60;Take one tablet twice a day with meals;12.50  
 Diclofenac;Inflammation;50;20;Take one tablet every 8 hours as needed;6.50  
 Pantoprazole;Acid Reflux;40;14;Take one tablet daily before breakfast;9.00  
 Salbutamol;Asthma;2;1;Inhale 2 puffs as needed up to 4 times daily;12.00  
 Prednisone;Inflammation;20;10;Take one tablet daily or as directed by a doctor;5.50  
 Omeprazole;Acid Reflux;20;14;Take one capsule daily before meals;8.00  
 Ranitidine;Acid Reflux;150;20;Take one tablet twice a day;7.50  
 Hydroxyzine;Allergy;25;30;Take one tablet 2-4 times daily;10.00  
 Zinc Supplement;Vitamin;50;30;Take one tablet daily;6.00

## 2.0 Distibution of work among team members

### 1. Muhammad Akmal Fikri Bin Asma Zaki

- Design and implementation of the Medicine class
- Development of accessor, mutator, and processing methods for medicine attributes
- Testing the functionality of the Medicine class

2. Wan Muhammad Hanif Bin Wan Muhammad Jaafar

- Implementation of LinkedList class
- Create main class and all processes using Linked Lists
- Testing the functionality of the Linked Lists main class

3. Muhammad Qais Qaiman Bin Mohd Nafiah

- Implementation of Queue class
- Create main class and all processes using Queue
- Testing the functionality of the Queue main class

4. Muhammad Amsyar Bin Jamaluddin

- Integration of the project components (classes and methods).
- Handling file I/O operations for reading and writing medicine data.
- Report generation and presentation preparation.

### 3.0 Complete Coding according by Classes

#### 3.1 Class Medicine

```
public class Medicine{  
    private String medicineName;  
    private String medicineType;  
    private double medicineDosage;  
    private int medicineQuantity;  
    private String medicineInstruction;
```

```
private double medicinePrice;

public Medicine(String medicineName, String medicineType, double
medicineDosage, int medicineQuantity, String medicineInstruction, double
medicinePrice){

    this.medicineName = medicineName;

    this.medicineType = medicineType;

    this.medicineDosage = medicineDosage;

    this.medicineQuantity = medicineQuantity;

    this.medicineInstruction = medicineInstruction;

    this.medicinePrice = medicinePrice;

}

public void setMedicineName(String medicineName) {

    this.medicineName = medicineName;

}

public void setMedicineType(String medicineType) {

    this.medicineType = medicineType;

}

public void setMedicineDosage(double medicineDosage) {

    this.medicineDosage = medicineDosage;

}

public void setMedicineQuantity(int medicineQuantity) {

    this.medicineQuantity = medicineQuantity;

}
```

```
public void setMedicineInstruction(String medicineInstruction) {  
    this.medicineInstruction = medicineInstruction;  
}
```

```
public void setMedicinePrice(double medicinePrice) {  
    this.medicinePrice = medicinePrice;  
}
```

```
public String getMedicineName() {  
    return medicineName;  
}
```

```
public String getMedicineType() {  
    return medicineType;  
}
```

```
public double getMedicineDosage() {  
    return medicineDosage;  
}
```

```
public int getMedicineQuantity() {  
    return medicineQuantity;  
}
```

```
public String getMedicineInstruction() {  
    return medicineInstruction;  
}
```



```

public double getMedicinePrice() {
    return medicinePrice;
}

public String toString(){
    return "Name : " + medicineName +
        "\nType : " + medicineType +
        "\nDosage : " + medicineDosage +
        "\nQuantity : " + medicineQuantity +
        "\nInstruction : " + medicineInstruction +
        "\nPrice : " + medicinePrice;
}

public String toStringToFile() {
    return medicineName + ";" + medicineType + ";" + medicineDosage + ";" +
        medicineQuantity + ";" + medicineInstruction + ";" + medicinePrice;
}
}

```

### 3.2 Class Linked List

```
public class LinkedList<E> {

    private Node<E> head, current, tail;

    public LinkedList() {
        head = current = tail = null;
    }

    public boolean isEmpty() {
        return head == null;
    }

    public void addFirst(E data) {
        Node newNode = new Node(data);
        newNode.next = this.head;
        this.head = newNode;
        if(this.tail == null) {
            this.tail = this.head;
        }
    }

    public void addLast(E data) {
        Node newNode = new Node(data);

        if(this.tail == null) {
            this.head = this.tail = newNode;
        }
    }
}
```

```
    }  
    else {  
        this.tail.next = newNode;  
        this.tail = this.tail.next;  
    }  
}
```

```
public E getFirst() {  
    if (this.isEmpty()) {  
        return null;  
    }  
    else {  
        this.current = this.head;  
        return this.current.data;  
    }  
}
```

```
public E getLast() {  
    if (this.isEmpty()) {  
        return null;  
    }  
    else {  
        return this.tail.data;  
    }  
}
```

```
public E getNext() {
```

```

    if (this.current == this.tail) {
        return null;
    }
    else {
        this.current = this.current.next;
        return this.current.data;
    }
}

public void clear() {
    this.head = this.current = this.tail = null;
}

public boolean contains(E data) {
    boolean isContain = false;
    this.current = this.head;

    while (this.current != null) {
        if (data.equals(this.current.data)) {
            isContain = true;
            break;
        }
    }

    return isContain;
}

```

```

public E removeFirst() {
    if (this.isEmpty()) {
        return null;
    }
    else {
        this.current = this.head;
        this.head = this.head.next;
        if (this.head == null)
            this.tail = null;
        return current.data;
    }
}

public E removeLast() {
    if (this.isEmpty())
        return null;
    else if (this.head == this.tail) {
        this.current = this.head;
        this.head = this.tail = null;
        return current.data;
    }
    else {
        this.current = this.head;
        while (this.current.next != tail) {
            this.current = this.current.next;
        }
    }
}

```

```

        Node<E> temp = this.tail;

        this.tail = this.current;

        this.tail.next = null;

        return temp.data;
    }
}

public E removeAfter(E data) {
    if (this.isEmpty()) {
        return null;
    }
    else if (this.head == this.tail) {
        this.current = this.head;
        this.head = this.tail = null;
        return current.data;
    }
    else {
        Node<E> previous = this.head;
        while (previous.next != null) {
            if (data.equals(previous.data))
            {
                break;
            }
            previous = previous.next;
        }
        current = previous.next;
    }
}

```

```

        previous.next = current.next;
        return current.data;
    }
}

public String toString() {
    StringBuilder result = new StringBuilder("[");
    if (this.isEmpty()) {
        result.append("The list is empty]");
    }
    else {
        this.current = this.head;
        while (this.current != null) {
            result.append(this.current.data);
            this.current = this.current.next;
            if (this.current != null)
                result.append(", ");
            else
                result.append("]");
        }
    }
    return result.toString();
}
}

```

### 3.3 Class Queue

```
public class Queue <E>
{
    private LinkedList <E> list;
    public Queue() {list = new LinkedList<E>();}

    public void enqueue(E data) {
        list.addLast(data);
    }

    public E dequeue() {
        return list.removeFirst();
    }

    public E getFront() {
        return list.getFirst();
    }

    public boolean isEmpty() {
        return list.isEmpty();
    }
}
```

### 3.4 Class Node

```
public class Node<E>{
    E data;
    Node next;

    public Node(E data) {
        this.data = data;
    }
}
```

### 3.5 Class MedicineMainLinkedList



```
import java.util.Scanner;
import java.io.*;
import java.util.*;

public class MedicineMainLinkedList
{
    public static void main(String[]args) throws IOException {

        LinkedList MedicineLL = new LinkedList();

        File input = new File("MedicineData.txt");

        try {
            Scanner in = new Scanner(input);
            while (in.hasNextLine()) {
                String s = in.nextLine();
                StringTokenizer st = new StringTokenizer(s, ";");

                String name = st.nextToken();
                String type = st.nextToken();
                double dosage = Double.parseDouble(st.nextToken());
                int quantity = Integer.parseInt(st.nextToken());
                String instruction = st.nextToken();
                double price = Double.parseDouble(st.nextToken());
```

```

        MedicineLL.addLast(new Medicine (name , type , dosage , quantity ,
instruction , price));

    }

} catch (Exception e) {

    System.out.println(e);

}


System.out.println("\nList of Medicine Data (LinkedList):");

System.out.println("\n=====");

Object obj = MedicineLL.getFirst();

while(obj != null)

{

    Medicine m = (Medicine) obj;

    System.out.println(m.toString() + "\n");

    obj = MedicineLL.getNext();

}


// Proses 1: search medicine quantity below 10

System.out.println("\nList of Medicine (Quantity < 10)");

System.out.println("\n=====");

obj = MedicineLL.getFirst();

while(obj != null)

{

    Medicine m = (Medicine) obj;

    if (m.getMedicineQuantity() <10){

        System.out.println(m.toString() + "\n");

    }

}

```

```

    obj = MedicineLL.getNext();
}

// Process 2 : search and calcalute price for medicine inventory
System.out.println("\nCalculate inventory Price for Paracetamol");
System.out.println("\n=====");
obj = MedicineLL.getFirst();
while(obj != null)
{
    Medicine m = (Medicine) obj;
    if (m.getMedicineName().equals("Paracetamol")){
        System.out.println(m.toString() + "\n");
        double totalPrice = m.getMedicineQuantity() * m.getMedicinePrice();
        System.out.println("Total Price : RM" + totalPrice);
    }
    obj = MedicineLL.getNext();
}

// Process 3 : calcalute total price for all inventory
System.out.println("\nCalculate inventory Price for Inventory");
System.out.println("\n=====");
double price = 0;
double totalPrice = 0;
obj = MedicineLL.getFirst();
while(obj != null)
{
    Medicine m = (Medicine) obj;

```

```

        price = m.getMedicineQuantity() * m.getMedicinePrice();

        totalPrice += price;

        obj = MedicineLL.getNext();
    }

    System.out.println("Total Price : RM" + totalPrice);

// Process 4 : Seperate the type of Medicine (using copy methode)
LinkedList FeverLL = new LinkedList();
LinkedList CoughLL = new LinkedList();
LinkedList InfectionLL = new LinkedList();
LinkedList AllergyLL = new LinkedList();
LinkedList VitaminLL = new LinkedList();
LinkedList OtherLL = new LinkedList();

while (!MedicineLL.isEmpty()){
    obj = MedicineLL.removeFirst();
    Medicine m = (Medicine) obj;
    if (m.getMedicineType().equals("Fever")){
        FeverLL.addLast(m);
    }
    else if (m.getMedicineType().equals("Cough")){
        CoughLL.addLast(m);
    }
    else if (m.getMedicineType().equals("Infection")){
        InfectionLL.addLast(m);
    }
    else if (m.getMedicineType().equals("Allergy")){

```

```

        AllergyLL.addLast(m);
    }
    else if(m.getMedicineType().equals("Vitamin")){
        VitaminLL.addLast(m);
    }
    else{
        OtherLL.addLast(m);
    }
}

//display what in FeverLL
System.out.println("\nList of FeverLL:");
System.out.println("\n=====");
obj = FeverLL.getFirst();
while(obj != null)
{
    Medicine m = (Medicine) obj;
    System.out.println(m.toString() + "\n");
    obj = FeverLL.getNext();
}

//display what in CoughLL
System.out.println("\nList of CoughLL:");
System.out.println("\n=====");
obj = CoughLL.getFirst();
while(obj != null)

```

```

{
    Medicine m = (Medicine) obj;
    System.out.println(m.toString() + "\n");
    obj = CoughLL.getNext();
}

//display what in InfectionLL
System.out.println("\nList of InfectionLL:");
System.out.println("\n=====");
obj = InfectionLL.getFirst();
while(obj != null)
{
    Medicine m = (Medicine) obj;
    System.out.println(m.toString() + "\n");
    obj = InfectionLL.getNext();
}

//display what in AllergyLL
System.out.println("\nList of AllergyLL:");
System.out.println("\n=====");
obj = AllergyLL.getFirst();
while(obj != null)
{
    Medicine m = (Medicine) obj;
    System.out.println(m.toString() + "\n");
    obj = AllergyLL.getNext();
}

```

```

//display what in VitaminLL
System.out.println("\nList of VitaminLL:");
System.out.println("\n=====");
obj = VitaminLL.getFirst();
while(obj != null)
{
    Medicine m = (Medicine) obj;
    System.out.println(m.toString() + "\n");
    obj = VitaminLL.getNext();
}

```

```

//display what in OtherLL
System.out.println("\nList of OtherLL:");
System.out.println("\n=====");
obj = OtherLL.getFirst();
while(obj != null)
{
    Medicine m = (Medicine) obj;
    System.out.println(m.toString() + "\n");
    obj = OtherLL.getNext();
}

```

```

// Process 5 : Search and update the medicine quantity base on user input
// and store at UpdatedData.txt

```

```

Scanner c = new Scanner(System.in);

System.out.println("Enter medicine name in FeverLL :");
String searchName = c.nextLine();


System.out.println("Enter the update quantity :");
int q = c.nextInt();


PrintWriter pw = new PrintWriter(new FileWriter("UpdatedData.txt"));


boolean found = false;


obj = FeverLL.getFirst();
while(obj != null)
{
    Medicine m = (Medicine) obj;
    if(m.getMedicineName().equalsIgnoreCase(searchName)){
        m.setMedicineQuantity(q);
        pw.write(m.toStringToFile());
        found = true;
        break;
    }
    obj = FeverLL.getNext();
}

if(found){

```



```

        System.out.println("\nData has been updated ");
        System.out.println("\n=====");
        obj = FeverLL.getFirst();
        while(obj != null)
        {
            Medicine m = (Medicine) obj;
            if(m.getMedicineName().equalsIgnoreCase(searchName)){
                System.out.println(m.toString() + "\n");
            }
            obj = FeverLL.getNext();
        }
    }
    else{
        System.out.println("Data Not Found");
    }

    pw.close();
}
}

```

### 3.6 Class MedicineMainQueue

```

import java.util.Scanner;
import java.io.*;
import java.util.*;

public class MedicineMainQueue
{

```

```

public static void main(String[]args) throws IOException {

    Queue qMedicine = new Queue();

    File input = new File("MedicineData.txt");

    try {
        Scanner in = new Scanner(input);
        while (in.hasNextLine()) {
            String s = in.nextLine();
            StringTokenizer st = new StringTokenizer(s, ",");

            String name = st.nextToken();
            String type = st.nextToken();
            double dosage = Double.parseDouble(st.nextToken());
            int quantity = Integer.parseInt(st.nextToken());
            String instruction = st.nextToken();
            double price = Double.parseDouble(st.nextToken());

            qMedicine.enqueue(new Medicine(name , type , dosage , quantity ,
instruction , price));
        }
    } catch (Exception e) {
        System.out.println(e);
    }

    Queue qTemp = new Queue();

    System.out.println("\nList of Medicine Data (Queue):");
    System.out.println("=====");
    while (!qMedicine.isEmpty()){
        Medicine m = (Medicine)qMedicine.dequeue();
        qTemp.enqueue(m);
        System.out.println(m.toString() + "\n");
    }
}

```

```

    }
    //move data in qTemp to original queue
    while(!qTemp.isEmpty()){
        qMedicine.enqueue(qTemp.dequeue());
    }

    // Proses 1: search medicine quantity below 10
    System.out.println("\nList of Medicine (Quantity < 10)");
    System.out.println("\n=====");
    while (!qMedicine.isEmpty()){
        Medicine m = (Medicine)qMedicine.dequeue();
        qTemp.enqueue(m);
        if (m.getMedicineQuantity() <10){
            System.out.println(m.toString() + "\n");
        }
    }
    while(!qTemp.isEmpty()){
        qMedicine.enqueue(qTemp.dequeue());
    }

    // Process 2 : search and calculate price for medicine inventory
    System.out.println("\nCalculate inventory Price for Paracetamol");
    System.out.println("\n=====");
    while (!qMedicine.isEmpty()){
        Medicine m = (Medicine)qMedicine.dequeue();
        qTemp.enqueue(m);
        if (m.getMedicineName().equals("Paracetamol")){
            System.out.println(m.toString() + "\n");
            double totalPrice = m.getMedicineQuantity() * m.getMedicinePrice();
            System.out.println("Total Price : RM" + totalPrice);
        }
    }
    while(!qTemp.isEmpty()){
        qMedicine.enqueue(qTemp.dequeue());
    }

```

```

}

// Process 3 : calculate total price for all inventory
System.out.println("\nCalculate inventory Price for Inventory");
System.out.println("\n=====");
double price = 0;
double totalPrice = 0;
while (!qMedicine.isEmpty()){
    Medicine m = (Medicine)qMedicine.dequeue();
    qTemp.enqueue(m);
    price = m.getMedicineQuantity() * m.getMedicinePrice();
    totalPrice += price;
}
System.out.println("Total Price : RM" + totalPrice);
while(!qTemp.isEmpty()){
    qMedicine.enqueue(qTemp.dequeue());
}

// Process 4 : Separate the type of Medicine (using copy method)
Queue qFever = new Queue();
Queue qCough = new Queue();
Queue qInfection = new Queue();
Queue qAllergy = new Queue();
Queue qVitamin = new Queue();
Queue qOther = new Queue();

while (!qMedicine.isEmpty()){
    Medicine m = (Medicine)qMedicine.dequeue();
    if (m.getMedicineType().equals("Fever")){
        qFever.enqueue(m);
    }
    else if (m.getMedicineType().equals("Cough")){
        qCough.enqueue(m);
    }
}

```

```

        else if (m.getMedicineType().equals("Infection")){
            qInfection.enqueue(m);
        }
        else if (m.getMedicineType().equals("Allergy")){
            qAllergy.enqueue(m);
        }
        else if(m.getMedicineType().equals("Vitamin")){
            qVitamin.enqueue(m);
        }
        else{
            qOther.enqueue(m);
        }
    }

    //display what in qFever
    System.out.println("\nList of qFever:");
    System.out.println("\n=====");
    while (!qFever.isEmpty()){
        Medicine m = (Medicine)qFever.dequeue();
        qTemp.enqueue(m);
        System.out.println(m.toString() + "\n");
    }
    while(!qTemp.isEmpty()){
        qFever.enqueue(qTemp.dequeue());
    }

    //display what in qCough
    System.out.println("\nList of qCough:");
    System.out.println("\n=====");
    while (!qCough.isEmpty()){
        Medicine m = (Medicine)qCough.dequeue();
        qTemp.enqueue(m);
        System.out.println(m.toString() + "\n");
    }

```

```

}
while(!qTemp.isEmpty()){
    qCough.enqueue(qTemp.dequeue());
}

//display what in qInfection
System.out.println("\nList of qInfection:");
System.out.println("\n=====");
while (!qInfection.isEmpty()){
    Medicine m = (Medicine)qInfection.dequeue();
    qTemp.enqueue(m);
    System.out.println(m.toString() + "\n");
}
while(!qTemp.isEmpty()){
    qInfection.enqueue(qTemp.dequeue());
}

//display what in qAllergy
System.out.println("\nList of qAllergy:");
System.out.println("\n=====");
while (!qAllergy.isEmpty()){
    Medicine m = (Medicine)qAllergy.dequeue();
    qTemp.enqueue(m);
    System.out.println(m.toString() + "\n");
}
while(!qTemp.isEmpty()){
    qAllergy.enqueue(qTemp.dequeue());
}

//display what in qVitamin
System.out.println("\nList of qVitamin:");
System.out.println("\n=====");
while (!qVitamin.isEmpty()){
    Medicine m = (Medicine)qVitamin.dequeue();

```

```

        qTemp.enqueue(m);
        System.out.println(m.toString() + "\n");
    }
    while(!qTemp.isEmpty()){
        qVitamin.enqueue(qTemp.dequeue());
    }

    //display qOther
    System.out.println("\nList of qOther");
    System.out.println("\n=====");
    while (!qOther.isEmpty()){
        Medicine m = (Medicine)qOther.dequeue();
        qTemp.enqueue(m);
        System.out.println(m.toString() + "\n");
    }
    while(!qTemp.isEmpty()){
        qOther.enqueue(qTemp.dequeue());
    }

    //Process 5 : Search and update the medicine quantity base on user input
    Scanner c = new Scanner(System.in);

    System.out.println("Enter medicine name in qFever :");
    String searchName = c.nextLine();

    System.out.println("Enter the update quantity :");
    int q = c.nextInt();

    PrintWriter pw = new PrintWriter(new FileWriter("UpdatedData.txt"));

    boolean found = false;

    while(!qFever.isEmpty())
    {

```

```

        Medicine m = (Medicine)qFever.dequeue();
        qTemp.enqueue(m);
        if(m.getMedicineName().equalsIgnoreCase(searchName)){
            m.setMedicineQuantity(q);
            pw.write(m.toStringToFile());
            found = true;
            break;
        }
    }
    while(!qTemp.isEmpty()){
        qFever.enqueue(qTemp.dequeue());
    }

    if (found) {
        System.out.println("\nData has been updated ");
        System.out.println("\n=====");
        while (!qFever.isEmpty()) {
            Medicine m = (Medicine) qFever.dequeue();
            qTemp.enqueue(m);
            if(m.getMedicineName().equalsIgnoreCase(searchName)){
                System.out.println(m.toString() + "\n");
            }
        }
        while (!qTemp.isEmpty()) {
            qFever.enqueue(qTemp.dequeue());
        }
    } else {
        System.out.println("Data Not Found");
    }

    pw.close();
}
}

```



## 4.0 Sample Output

### 4.1 MedicineMainLinkedList

#### Process 1: Store the medicine data to file

List of Medicine Data (LinkedList):

=====

Name : Paracetamol

Type : Fever

Dosage : 500.0

Quantity : 10

Instruction : Take one tablet every 4-6 hours as needed

Price : 2.5

Name : Cetirizine

Type : Allergy

Dosage : 10.0

Quantity : 15

Instruction : Take one tablet daily

Price : 5.0

Name : Vitamin C

Type : Vitamin

Dosage : 500.0

Quantity : 20

Instruction : Take one tablet after meals daily

Price : 7.0

Name : Cough Syrup

Type : Cough

Dosage : 10.0

Quantity : 100

Instruction : Take 10 mL three times a day

Price : 4.5

Name : Ibuprofen

Type : Fever

Dosage : 200.0

Quantity : 12

Instruction : Take one tablet every 6-8 hours as needed

Price : 3.0

Name : Amoxicillin  
Type : Infection  
Dosage : 500.0  
Quantity : 14  
Instruction : Take one capsule every 8 hours for 7 days  
Price : 10.0

Name : Loratadine  
Type : Allergy  
Dosage : 10.0  
Quantity : 10  
Instruction : Take one tablet daily  
Price : 4.0

Name : Azithromycin  
Type : Infection  
Dosage : 250.0  
Quantity : 6  
Instruction : Take two tablets on day 1 then one tablet daily for 4 days  
Price : 12.0

Name : Multivitamin  
Type : Vitamin  
Dosage : 1.0  
Quantity : 30  
Instruction : Take one tablet daily with meals  
Price : 8.5

Name : Dextromethorphan  
Type : Cough  
Dosage : 15.0  
Quantity : 120  
Instruction : Take 15 mL every 6 hours as needed  
Price : 6.0

Name : Losartan  
Type : Blood Pressure  
Dosage : 50.0  
Quantity : 30  
Instruction : Take one tablet daily  
Price : 15.0

Name : Metformin  
Type : Diabetes  
Dosage : 500.0  
Quantity : 60  
Instruction : Take one tablet twice a day with meals  
Price : 12.5

Name : Diclofenac  
Type : Inflammation  
Dosage : 50.0  
Quantity : 20  
Instruction : Take one tablet every 8 hours as needed  
Price : 6.5

Name : Pantoprazole  
Type : Acid Reflux  
Dosage : 40.0  
Quantity : 14  
Instruction : Take one tablet daily before breakfast  
Price : 9.0

Name : Salbutamol  
Type : Asthma  
Dosage : 2.0  
Quantity : 1  
Instruction : Inhale 2 puffs as needed up to 4 times daily  
Price : 12.0

Name : Prednisone  
Type : Inflammation  
Dosage : 20.0  
Quantity : 10  
Instruction : Take one tablet daily or as directed by a doctor  
Price : 5.5

Name : Omeprazole  
Type : Acid Reflux  
Dosage : 20.0  
Quantity : 14  
Instruction : Take one capsule daily before meals  
Price : 8.0

Name : Ranitidine  
Type : Acid Reflux  
Dosage : 150.0  
Quantity : 20  
Instruction : Take one tablet twice a day  
Price : 7.5

Name : Hydroxyzine  
Type : Allergy  
Dosage : 25.0  
Quantity : 30  
Instruction : Take one tablet 2-4 times daily  
Price : 10.0

Name : Zinc Supplement  
Type : Vitamin  
Dosage : 50.0  
Quantity : 30  
Instruction : Take one tablet daily  
Price : 6.0

## Proses 2: Search medicine quantity below 10.

List of Medicine (Quantity < 10)

=====

Name : Azithromycin  
Type : Infection  
Dosage : 250.0  
Quantity : 6  
Instruction : Take two tablets on day 1 then one tablet daily for 4 days  
Price : 12.0

Name : Salbutamol  
Type : Asthma  
Dosage : 2.0  
Quantity : 1  
Instruction : Inhale 2 puffs as needed up to 4 times daily  
Price : 12.0

**Proses 3 : Calculate the total price medicine for specific medicine.**

```
Calculate inventory Price for Paracetamol

=====
Name : Paracetamol
Type : Fever
Dosage : 500.0
Quantity : 10
Instruction : Take one tablet every 4-6 hours as needed
Price : 2.5

Total Price : RM25.0
```

**Proses 4: Calculate the total price of all medicine inventory**

```
Calculate inventory Price for Inventory

=====
Total Price : RM4218.0
```

**Proses 5 : Split the medicine data according medicine type**

```
List of FeverLL:

=====
Name : Paracetamol
Type : Fever
Dosage : 500.0
Quantity : 10
Instruction : Take one tablet every 4-6 hours as needed
Price : 2.5

Name : Ibuprofen
Type : Fever
Dosage : 200.0
Quantity : 12
Instruction : Take one tablet every 6-8 hours as needed
Price : 3.0
```

List of CoughLL:

=====

Name : Cough Syrup

Type : Cough

Dosage : 10.0

Quantity : 100

Instruction : Take 10 mL three times a day

Price : 4.5

Name : Dextromethorphan

Type : Cough

Dosage : 15.0

Quantity : 120

Instruction : Take 15 mL every 6 hours as needed

Price : 6.0

List of InfectionLL:

=====

Name : Amoxicillin

Type : Infection

Dosage : 500.0

Quantity : 14

Instruction : Take one capsule every 8 hours for 7 days

Price : 10.0

Name : Azithromycin

Type : Infection

Dosage : 250.0

Quantity : 6

Instruction : Take two tablets on day 1 then one tablet daily for 4 days

Price : 12.0

List of AllergyLL:

=====

Name : Cetirizine

Type : Allergy

Dosage : 10.0

Quantity : 15

Instruction : Take one tablet daily

Price : 5.0

Name : Loratadine

Type : Allergy

Dosage : 10.0

Quantity : 10

Instruction : Take one tablet daily

Price : 4.0

Name : Hydroxyzine

Type : Allergy

Dosage : 25.0

Quantity : 30

Instruction : Take one tablet 2-4 times daily

Price : 10.0

List of VitaminLL:

=====

Name : Vitamin C

Type : Vitamin

Dosage : 500.0

Quantity : 20

Instruction : Take one tablet after meals daily

Price : 7.0

Name : Multivitamin

Type : Vitamin

Dosage : 1.0

Quantity : 30

Instruction : Take one tablet daily with meals

Price : 8.5

Name : Zinc Supplement

Type : Vitamin

Dosage : 50.0

Quantity : 30

Instruction : Take one tablet daily

Price : 6.0



List of OtherLL:

=====

Name : Losartan  
Type : Blood Pressure  
Dosage : 50.0  
Quantity : 30  
Instruction : Take one tablet daily  
Price : 15.0

Name : Metformin  
Type : Diabetes  
Dosage : 500.0  
Quantity : 60  
Instruction : Take one tablet twice a day with meals  
Price : 12.5

Name : Diclofenac  
Type : Inflammation  
Dosage : 50.0  
Quantity : 20  
Instruction : Take one tablet every 8 hours as needed  
Price : 6.5

Name : Pantoprazole  
Type : Acid Reflux  
Dosage : 40.0  
Quantity : 14  
Instruction : Take one tablet daily before breakfast  
Price : 9.0

Name : Salbutamol  
Type : Asthma  
Dosage : 2.0  
Quantity : 1  
Instruction : Inhale 2 puffs as needed up to 4 times daily  
Price : 12.0

Name : Prednisone  
Type : Inflammation  
Dosage : 20.0  
Quantity : 10  
Instruction : Take one tablet daily or as directed by a doctor  
Price : 5.5

Name : Omeprazole  
Type : Acid Reflux  
Dosage : 20.0  
Quantity : 14  
Instruction : Take one capsule daily before meals  
Price : 8.0

Name : Ranitidine  
Type : Acid Reflux  
Dosage : 150.0  
Quantity : 20  
Instruction : Take one tablet twice a day  
Price : 7.5

### Process 6: Search and update the medicine quantity base on user input and store at UpdatedData.txt

```
Enter medicine name in FeverLL :  
paracetamol  
Enter the update quantity :  
20  
  
Data has been updated  
  
=====  
Name : Paracetamol  
Type : Fever  
Dosage : 500.0  
Quantity : 20  
Instruction : Take one tablet every 4-6 hours as needed  
Price : 2.5
```

## 4.2 MedicineMainQueue

### Process 1: Store the medicine data to file

List of Medicine Data (Queue):

=====

Name : Paracetamol

Type : Fever

Dosage : 500.0

Quantity : 10

Instruction : Take one tablet every 4-6 hours as needed

Price : 2.5

Name : Cetirizine

Type : Allergy

Dosage : 10.0

Quantity : 15

Instruction : Take one tablet daily

Price : 5.0

Name : Vitamin C

Type : Vitamin

Dosage : 500.0

Quantity : 20

Instruction : Take one tablet after meals daily

Price : 7.0

Name : Cough Syrup

Type : Cough

Dosage : 10.0

Quantity : 100

Instruction : Take 10 mL three times a day

Price : 4.5

Name : Ibuprofen

Type : Fever

Dosage : 200.0

Quantity : 12

Instruction : Take one tablet every 6-8 hours as needed

Price : 3.0

Name : Amoxicillin

Type : Infection

Dosage : 500.0

Quantity : 14

Instruction : Take one capsule every 8 hours for 7 days

Price : 10.0

Name : Loratadine

Type : Allergy

Dosage : 10.0

Quantity : 10

Instruction : Take one tablet daily

Price : 4.0

Name : Azithromycin

Type : Infection

Dosage : 250.0

Quantity : 6

Instruction : Take two tablets on day 1 then one tablet daily for 4 days

Price : 12.0

Name : Multivitamin

Type : Vitamin

Dosage : 1.0

Quantity : 30

Instruction : Take one tablet daily with meals

Price : 8.5

Name : Dextromethorphan

Type : Cough

Dosage : 15.0

Quantity : 120

Instruction : Take 15 mL every 6 hours as needed

Price : 6.0

Name : Losartan

Type : Blood Pressure

Dosage : 50.0

Quantity : 30

Instruction : Take one tablet daily

Price : 15.0

Name : Metformin

Type : Diabetes

Dosage : 500.0

Quantity : 60

Instruction : Take one tablet twice a day with meals

Price : 12.5

Name : Diclofenac  
Type : Inflammation  
Dosage : 50.0  
Quantity : 20  
Instruction : Take one tablet every 8 hours as needed  
Price : 6.5

Name : Pantoprazole  
Type : Acid Reflux  
Dosage : 40.0  
Quantity : 14  
Instruction : Take one tablet daily before breakfast  
Price : 9.0

Name : Salbutamol  
Type : Asthma  
Dosage : 2.0  
Quantity : 1  
Instruction : Inhale 2 puffs as needed up to 4 times daily  
Price : 12.0

Name : Prednisone  
Type : Inflammation  
Dosage : 20.0  
Quantity : 10  
Instruction : Take one tablet daily or as directed by a doctor  
Price : 5.5

Name : Omeprazole  
Type : Acid Reflux  
Dosage : 20.0  
Quantity : 14  
Instruction : Take one capsule daily before meals  
Price : 8.0

Name : Ranitidine  
Type : Acid Reflux  
Dosage : 150.0  
Quantity : 20  
Instruction : Take one tablet twice a day  
Price : 7.5

Name : Hydroxyzine  
Type : Allergy  
Dosage : 25.0  
Quantity : 30  
Instruction : Take one tablet 2-4 times daily  
Price : 10.0

Name : Zinc Supplement  
Type : Vitamin  
Dosage : 50.0  
Quantity : 30  
Instruction : Take one tablet daily  
Price : 6.0

## Process 2: Search medicine quantity below 10.

List of Medicine (Quantity < 10)

=====

Name : Azithromycin

Type : Infection

Dosage : 250.0

Quantity : 6

Instruction : Take two tablets on day 1 then one tablet daily for 4 days

Price : 12.0

Name : Salbutamol

Type : Asthma

Dosage : 2.0

Quantity : 1

Instruction : Inhale 2 puffs as needed up to 4 times daily

Price : 12.0

## Process 3: Calculate the total price medicine for specific medicine

Calculate inventory Price for Paracetamol

=====

Name : Paracetamol

Type : Fever

Dosage : 500.0

Quantity : 10

Instruction : Take one tablet every 4-6 hours as needed

Price : 2.5

Total Price : RM25.0

## Process 4: Calculate the total price of all medicine inventory

Calculate inventory Price for Inventory

=====

Total Price : RM4218.0

**Process 5: Split the medicine data according medicine type.**

List of qFever:

=====

Name : Paracetamol

Type : Fever

Dosage : 500.0

Quantity : 10

Instruction : Take one tablet every 4-6 hours as needed

Price : 2.5

Name : Ibuprofen

Type : Fever

Dosage : 200.0

Quantity : 12

Instruction : Take one tablet every 6-8 hours as needed

Price : 3.0

List of qCough:

=====

Name : Cough Syrup

Type : Cough

Dosage : 10.0

Quantity : 100

Instruction : Take 10 mL three times a day

Price : 4.5

Name : Dextromethorphan

Type : Cough

Dosage : 15.0

Quantity : 120

Instruction : Take 15 mL every 6 hours as needed

Price : 6.0

List of qInfection:

=====

Name : Amoxicillin

Type : Infection

Dosage : 500.0

Quantity : 14

Instruction : Take one capsule every 8 hours for 7 days

Price : 10.0

Name : Azithromycin

Type : Infection

Dosage : 250.0

Quantity : 6

Instruction : Take two tablets on day 1 then one tablet daily for 4 days

Price : 12.0

List of qAllergy:

=====

Name : Cetirizine

Type : Allergy

Dosage : 10.0

Quantity : 15

Instruction : Take one tablet daily

Price : 5.0

Name : Loratadine

Type : Allergy

Dosage : 10.0

Quantity : 10

Instruction : Take one tablet daily

Price : 4.0

Name : Hydroxyzine

Type : Allergy

Dosage : 25.0

Quantity : 30

Instruction : Take one tablet 2-4 times daily

Price : 10.0



List of qVitamin:

=====

Name : Vitamin C

Type : Vitamin

Dosage : 500.0

Quantity : 20

Instruction : Take one tablet after meals daily

Price : 7.0

Name : Multivitamin

Type : Vitamin

Dosage : 1.0

Quantity : 30

Instruction : Take one tablet daily with meals

Price : 8.5

Name : Zinc Supplement

Type : Vitamin

Dosage : 50.0

Quantity : 30

Instruction : Take one tablet daily

Price : 6.0

List of qOther

=====

Name : Losartan  
Type : Blood Pressure  
Dosage : 50.0  
Quantity : 30  
Instruction : Take one tablet daily  
Price : 15.0

Name : Metformin  
Type : Diabetes  
Dosage : 500.0  
Quantity : 60  
Instruction : Take one tablet twice a day with meals  
Price : 12.5

Name : Diclofenac  
Type : Inflammation  
Dosage : 50.0  
Quantity : 20  
Instruction : Take one tablet every 8 hours as needed  
Price : 6.5

Name : Pantoprazole  
Type : Acid Reflux  
Dosage : 40.0  
Quantity : 14  
Instruction : Take one tablet daily before breakfast  
Price : 9.0

Name : Salbutamol  
Type : Asthma  
Dosage : 2.0  
Quantity : 1  
Instruction : Inhale 2 puffs as needed up to 4 times daily  
Price : 12.0

Name : Prednisone  
Type : Inflammation  
Dosage : 20.0  
Quantity : 10  
Instruction : Take one tablet daily or as directed by a doctor  
Price : 5.5

Name : Omeprazole  
Type : Acid Reflux  
Dosage : 20.0  
Quantity : 14  
Instruction : Take one capsule daily before meals  
Price : 8.0

Name : Ranitidine  
Type : Acid Reflux  
Dosage : 150.0  
Quantity : 20  
Instruction : Take one tablet twice a day  
Price : 7.5

**Process 6: Search and update the medicine quantity base on user input and store at UpdatedData.txt**

Enter medicine name in qFever :

paracetamol

Enter the update quantity :

20

Data has been updated

=====

Name : Paracetamol

Type : Fever

Dosage : 500.0

Quantity : 20

Instruction : Take one tablet every 4-6 hours as needed

Price : 2.5

## 5.0 Conclusion

Based on the requirements of Dr. Azhar Healthcare Pharmacy's application, using a LinkedList would be the most appropriate choice for managing medicine inventory and transactions. A LinkedList offers several advantages over a Queue, including easier and more efficient access to data, direct iteration, and flexible updates. Specifically, tasks such as searching for medicines with low quantities, updating quantities, calculating prices, and splitting medicines by type are all more straightforward and efficient with a LinkedList. It allows for direct traversal and manipulation of data, without the need for dequeuing and re-enqueueing operations, as would be required in a Queue.

Additionally, LinkedList simplifies code implementation and maintenance, making it a better choice for handling the dynamic nature of the pharmacy's inventory system. Its ability to easily handle updates, search operations, and report generation ensures that the application can scale and provide better operational efficiency. Therefore, a LinkedList is the most suitable data structure for this pharmacy application.