



SCHOOL OF COMPUTING (SOC)
SECOND SEMESTER SESSION 2022/2023 (A222)
STIA 2024 DATA STRUCTURE AND ALGORITHM ANALYSIS

GROUP B

GROUP PROJECT

PREPARED FOR:

PROF. MADYA DR. FAUZIAH BT BAHAROM

GROUP MEMBERS:

NAME	MATRIC NO.
LEONG MING YUAN	277504
LEE JUAN	280027
OH BOON JOOI	284925
CHEW KAI XIAN	285789

TABLE OF CONTENTS

Content	Page
SOURCE CODE (WITHOUT GUI)	2
Product.java	2
Customer.java.....	3
Node.java	4
LinkedListOperation.java.....	4
CompanyABCManagementSystem.java (Main class).....	7
SOURCE CODE (WITH GUI).....	15
Product.java	15
Customer.java.....	16
Node.java	16
CompanyABCManagementSystemGUI.java	17
SAMPLE OUTPUT (WITHOUT GUI).....	32
SAMPLE OUTPUT (WITH GUI).....	40
PRESENTATION VIDEO LINK	55

SOURCE CODE (WITHOUT GUI)

Attached below is the complete source code for the **console-based program (without GUI)**, separated into five parts that represent the five Java classes used:

- *Product.java*
- *Customer.java*
- *Node.java*
- *LinkedListOperation.java*
- *CompanyABCManagementSystem.java* (**Main class**)

--- **Product.java**

```
package dsgroupproject;

public class Product {

    private String productName;
    private int purchaseQuantity;
    private String purchaseDate;

    public Product(String productName, int purchaseQuantity, String purchaseDate)
    {
        this.productName = productName;
        this.purchaseQuantity = purchaseQuantity;
        this.purchaseDate = purchaseDate;
    }

    //no-arg constructor to allow PList ArrayList to pass through without
    arguments
    public Product() {
    }

    public String getProductName() {
        return productName;
    }

    public int getPurchaseQuantity() {
        return purchaseQuantity;
    }

    public String getPurchaseDate(){
        return purchaseDate;
    }

    public void setProductName(String productName) {
        this.productName = productName;
    }

    public void setPurchaseQuantity(int purchaseQuantity) {
        this.purchaseQuantity = purchaseQuantity;
    }
}
```

```

    public void setPurchaseDate(String purchaseDate) {
        this.purchaseDate = purchaseDate;
    }
}

```

--- Customer.java

```

package dsgroupproject;

import java.util.ArrayList;

public class Customer extends Product {

    private int id;
    private String name;
    private int phoneNumber;
    private ArrayList<Product> pList;

    //constructor for new customer object
    public Customer(int id, String name, int phoneNumber, ArrayList<Product>
pList) {
        this.id = id;
        this.name = name;
        this.phoneNumber = phoneNumber;
        this.pList = pList;
    }

    //constructor for updated customer personal info object
    public Customer(String name, int phoneNumber) {
        this.name = name;
        this.phoneNumber = phoneNumber;
    }

    //constructor for updated customer products info object
    public Customer(ArrayList<Product> pList) {
        this.pList = pList;
    }

    public int getCustomerID() {
        return id;
    }

    public String getCustomerName() {
        return name;
    }

    public int getCustomerphoneNumber() {
        return phoneNumber;
    }

    public ArrayList<Product> getPList() {
        return pList;
    }

    public void setCustomerName(String name) {
        this.name = name;
    }
}

```

```

    public void setCustomerphoneNumber(int phoneNumber) {
        this.phoneNumber = phoneNumber;
    }
}

```

--- Node.java

```

package dsgruppproject;

public class Node {

    private Customer info;
    private Node link;

    public Node() {
        this.info = null;
        this.link = null;
    }

    public Node(Customer newInfo) {
        this.info = newInfo;
        this.link = null;
    }

    public void setInfo(Customer a) {
        this.info = a;
    }

    public void setLink(Node b) {
        this.link = b;
    }

    public Customer getInfo() {
        return this.info;
    }

    public Node getLink() {
        return this.link;
    }
}

```

--- LinkedListOperation.java

```

package dsgruppproject;

import java.util.ArrayList;

public class LinkedListOperation {

    private Node head = null;
}

```

```

public void addData(Customer a) {
    Node newNode = new Node(a);
    if (head == null)
        head = newNode;
    else {
        Node current = head;
        while (current.getLink() != null) {
            current = current.getLink();
        }
        current.setLink(newNode);
    }
}

public void deleteTarget(int target) {
    if (head.getInfo().getCustomerID() == target) {
        head = head.getLink();
    }
    else {
        Node before = null;
        Node current = head;

        while ((current.getInfo().getCustomerID() != target) &&
(current.getLink() != null)) {
            before = current;
            current = current.getLink();
        }

        if (current.getInfo().getCustomerID() == target)
            before.setLink(current.getLink());
        else
            System.out.println("Error. There is nothing matches the item that
you wish to delete");
    }
}

public void displayCustomerList() {
    Node current = head;

    if (current == null)
        System.out.println("--> The list is empty, add new data!");
    else {
        System.out.printf("|%-3s|%-20s|%-15s|%-20s|%-10s|%-20s|\n", "ID.",
"CUSTOMER NAME", "PHONE NUMBER", "PRODUCT NAME", "QUANTITY", "PURCHASE DATE");
        while (current != null) {
            System.out.printf("|%-3d|", current.getInfo().getCustomerID());
            System.out.printf("%-20s|", current.getInfo().getCustomerName());
            System.out.printf("0%-14s|",
current.getInfo().getCustomerphoneNumber());

            ArrayList<Product> pList = current.getInfo().getPList();
            boolean firstProduct = true;

            for (Product product : pList) {
                if (firstProduct) {
                    System.out.printf("%20s|%-10d|%-20s|\n",
product.getProductname(), product.getPurchaseQuantity(),
product.getPurchaseDate());
                    firstProduct = false;
                }
                else {
                    System.out.printf("|%-3s|%-20s|%-15s|%-20s|%-10d|%-20s|\n",
"", "", "", product.getProductname(), product.getPurchaseQuantity(),
product.getPurchaseDate());
                }
            }

            current = current.getLink();
        }
    }
}

```

```

    }

}

public void search(int target) {
    Node current = head;
    int searchCount = 1;

    while ((current.getInfo().getCustomerID() != target) &&
(current.getLink() != null)) {
        current = current.getLink();
        searchCount++;
    }

    if (current.getInfo().getCustomerID() == target) {
        System.out.printf("|%-3s|%-20s|%-15s|%-20s|%-10s|%-20s|\n", "ID.",
"CUSTOMER NAME", "PHONE NUMBER", "PRODUCT NAME", "QUANTITY", "PURCHASE DATE");
        while (current != null) {
            System.out.printf("|%-3d|", current.getInfo().getCustomerID());
            System.out.printf("%-20s|", current.getInfo().getCustomerName());
            System.out.printf("0%-14s|",
current.getInfo().getCustomerphoneNumber());

            ArrayList<Product> pList = current.getInfo().getPList();
            boolean firstProduct = true;

            for (Product product : pList) {
                if (firstProduct) {
                    System.out.printf("%20s|%-10d|%-20s|\n",
product.getProductname(), product.getPurchaseQuantity(),
product.getPurchaseDate());
                    firstProduct = false;
                }
                else {
                    System.out.printf("|%-40s|%-20s|%-10d|%-20s|\n", "",
product.getProductname(), product.getPurchaseQuantity(),
product.getPurchaseDate());
                }
            }

            break;
        }
    }
    else
        System.out.println("There is nothing matches for your search.");
}

public boolean isTheTargetOnTheList(int target) {
    boolean search = false;

    if (head == null)
        System.out.println("\n--> The list is empty.");
    else {
        Node current = head;
        while (current.getInfo().getCustomerID() != target &&
current.getLink() != null) {
            current = current.getLink();
        }

        if (current.getInfo().getCustomerID() == target)
            search = true;
    }
    return search;
}

public void updateTarget(int target, Customer newInfo, int updateType) {
    if (head.getInfo().getCustomerID() == target) {
        if (updateType == 1) {

```

```

        head.getInfo().setCustomerName(newInfo.getCustomerName());
head.getInfo().setCustomerphoneNumber(newInfo.getCustomerphoneNumber());
    }
    else {
        ArrayList<Product> pList = head.getInfo().getPList();
        pList.clear();          //clear existing products

        ArrayList<Product> updatedpList = newInfo.getPList();
        pList.addAll(updatedpList);    //add updated products list
    }
}
else {
    Node current = head;

    while ((current.getInfo().getCustomerID() != target) &&
(current.getLink() != null)) {
        current = current.getLink();
    }
    if (current.getInfo().getCustomerID() == target) {
        if (updateType == 1) {
            current.getInfo().setCustomerName(newInfo.getCustomerName());

current.getInfo().setCustomerphoneNumber(newInfo.getCustomerphoneNumber());
        }
        else {
            ArrayList<Product> pList = current.getInfo().getPList();
            pList.clear();          // Clear the existing products

            ArrayList<Product> updatedpList = newInfo.getPList();
            pList.addAll(updatedpList);    //add updated products list
        }
    }
    else
        System.out.println("Error. There is nothing matches the item that
you wish to update");
}
}
}
}

```

--- CompanyABCManagementSystem.java (Main class)

```

package dsgrupproject;

import java.util.ArrayList;
import java.util.Scanner;
import java.util.LinkedList;
import java.util.InputMismatchException;

public class CompanyABCManagementSystem {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        int id = 1;
        int choice;
        LinkedListOperation customers = new LinkedListOperation();

        String customerName;
        int phoneNumber;
    }
}

```



```

int noOfProducts;
int productNo;
String pName;
String pDate;
int pQuantity;

while (true) {

    try {
        System.out.println("*****");
        System.out.println("Welcome To Company ABC\n");
        System.out.println("1. Display Data");
        System.out.println("2. Add Data");
        System.out.println("3. Search Data");
        System.out.println("4. Update Data");
        System.out.println("5. Delete Data");
        System.out.println("6. Exit");
        System.out.print("\nEnter your choice: ");
        choice = scanner.nextInt();
        scanner.nextLine(); // Consume newline character

        switch (choice) {

            //1. Display Data
            case 1:

                System.out.println("\n-----\nDISPLAY LIST\n");

                customers.displayCustomerList();

                System.out.print("\nEnter any number to return: ");
                choice = scanner.nextInt();
                System.out.println("");
                break;

            //2. Add Data
            case 2:

                System.out.println("\n-----\nADD DATA\n");
                System.out.print("Enter customer name: ");
                customerName = scanner.nextLine();

                while (true) {
                    try {
                        System.out.print("Enter customer phone number:
");

                        phoneNumber = scanner.nextInt();
                        scanner.nextLine();
                        break;
                    } catch (InputMismatchException imx) {
                        System.out.println("--> Phone number can only
have digits! Try again.");
                        scanner.nextLine();
                    }
                }

                while (true) {
                    try {
                        System.out.print("Enter number of products
purchased: ");

                        noOfProducts = scanner.nextInt();

                        if ((noOfProducts <= 0))
                            throw new Exception("--> Invalid digit! Try
again.");

                        scanner.nextLine();
                        break;

```

```

        } catch (InputMismatchException imx) {
            System.out.println("--> Insert digits only! Try
again.");

            scanner.nextLine();
        } catch (Exception x) {
            System.out.println(x.getMessage());
            scanner.nextLine();
        }
    }

    ArrayList<Product> pList = new ArrayList<>();

    for (int i=1; i<=noOfProducts; i++) {
        boolean cond = true;
        while (cond) {
            try {
                System.out.println("\nProduct " + i);
                System.out.println("-- 1. Air purifier");
                System.out.println("-- 2. Water purifier");
                System.out.println("-- 3. Air conditioner");
                System.out.print("Select product type for
Product " + i + ": ");

                productNo = scanner.nextInt();

                if ((productNo != 1) && (productNo != 2) &&
                    (productNo != 3))
                    throw new Exception("--> Invalid digit!
Try again.");

                if (productNo == 1)
                    pName = "Air purifier";
                else if (productNo == 2)
                    pName = "Water purifier";
                else
                    pName = "Air conditioner";

                scanner.nextLine();

                System.out.print("Enter purchase date: ");
                pDate = scanner.nextLine();

                while (true) {
                    try {
                        System.out.print("Enter quantity: ");
                        pQuantity = scanner.nextInt();

                        if (pQuantity <= 0)
                            throw new Exception("--> Invalid
quantity! Try again.");

                        break;
                    } catch (InputMismatchException imx) {
                        System.out.println("--> Quantity can
only have digits! Try again.");

                        scanner.nextLine();
                    } catch (Exception x) {
                        System.out.println(x.getMessage());
                    }
                }

                Product product = new Product(pName,
pQuantity, pDate);

                pList.add(product);

                break;
            } catch (InputMismatchException imx) {

```

```

        System.out.println("--> Enter digits only!
Try again.");
        scanner.nextLine();
    } catch (Exception x) {
        System.out.println(x.getMessage());
        scanner.nextLine();
    }
}

Customer customer = new Customer(id, customerName,
phoneNumber, pList);
id++;
customers.addData(customer);

System.out.println("\n--> Customer added successfully.");
System.out.print("\nEnter any number to return: ");
choice = scanner.nextInt();
break;

//3. Search Data
case 3:

    System.out.println("\n-----\nSEARCH DATA\n");

    System.out.print("Please enter Customer ID for Search :
");

    int searchID = scanner.nextInt();

    if (customers.isTheTargetOnTheList(searchID) == true) {
        System.out.println("");
        customers.search(searchID);
        System.out.print("\nEnter any number to return: ");
        choice = scanner.nextInt();
        break;
    }
    else
        System.out.println("--> There is no data that matches
your ID.");

    System.out.print("\nEnter any number to return: ");
    choice = scanner.nextInt();
    break;

//4. Update Data
case 4:

    int updateID;

    System.out.println("\n-----\nUPDATE DATA\n");
    System.out.print("Enter customer ID to update: ");
    updateID = scanner.nextInt();

    if (customers.isTheTargetOnTheList(updateID) == true) {
        scanner.nextLine();

        while (true) {
            try {
                System.out.println("\nSelect info to update
for customer ID " + updateID);
                System.out.println("-- 1. Update Personal
Info (Name, Phone Number)");
                System.out.println("-- 2. Update Products
Info");
                System.out.print("Enter number 1 or 2 to
update respective info, or any other number to cancel: ");
                choice = scanner.nextInt();

```

```

        break;
    } catch (InputMismatchException imx) {
        System.out.println("--> Insert digits only!
Try again.");
        scanner.nextLine();
    }
}

if ((choice != 1) && (choice != 2))
    break;

//user chooses to update personal info
if (choice == 1) {
    scanner.nextLine();
    System.out.print("Enter customer name: ");
    customerName = scanner.nextLine();

    while (true) {
        try {
            System.out.print("Enter customer phone
number: ");

            phoneNumber = scanner.nextInt();
            scanner.nextLine();
            break;
        } catch (InputMismatchException imx) {
            System.out.println("--> Phone number can
only have digits! Try again.");
            scanner.nextLine();
        }
    }

    Customer updateCustomer = new
Customer(customerName, phoneNumber);
    customers.updateTarget(updateID, updateCustomer,
1);

}

//if user chooses to update products info
else {
    scanner.nextLine();

    while (true) {
        try {
            System.out.print("Enter number of
products purchased: ");

            noOfProducts = scanner.nextInt();

            if ((noOfProducts <= 0))
                throw new Exception("--> Invalid
digit! Try again.");

            scanner.nextLine();
            break;
        } catch (InputMismatchException imx) {
            System.out.println("--> Insert digits
only! Try again.");
            scanner.nextLine();
        } catch (Exception x) {
            System.out.println(x.getMessage());
            scanner.nextLine();
        }
    }

    ArrayList<Product> updatedpList = new
ArrayList<>();

    for (int i=1; i<=noOfProducts; i++) {

```

```

boolean cond = true;
while (cond) {
    try {
        System.out.println("\nProduct " + i);
        System.out.println("-- 1. Air
purifier");
        System.out.println("-- 2. Water
purifier");
        System.out.println("-- 3. Air
conditioner");
        for Product " + i + ": ";
        2) && (productNo != 3))
        digit! Try again.");

        date: ";

        quantity: ";
        scanner.nextInt();

        Invalid quantity! Try again.");

        imx) {
            Quantity can only have digits! Try again.");

            System.out.println(x.getMessage());

            pQuantity, pDate);

            only! Try again.");

        boolean cond = true;
        while (cond) {
            try {
                System.out.println("\nProduct " + i);
                System.out.println("-- 1. Air
                purifier");
                System.out.println("-- 2. Water
                purifier");
                System.out.println("-- 3. Air
                conditioner");
                System.out.print("Select product type
                productNo = scanner.nextInt();

                if ((productNo != 1) && (productNo !=
                    throw new Exception("--> Invalid

                if (productNo == 1)
                    pName = "Air purifier";
                else if (productNo == 2)
                    pName = "Water purifier";
                else
                    pName = "Air conditioner";

                scanner.nextLine();

                System.out.print("Enter purchase
                pDate = scanner.nextLine();

                while (true) {
                    try {
                        System.out.print("Enter

                        pQuantity =

                        if (pQuantity <= 0)
                            throw new Exception("-->

                        break;

                    } catch (InputMismatchException

                        System.out.println("-->

                        scanner.nextLine();
                    } catch (Exception x) {

                        }

                Product product = new Product(pName,
                updatedpList.add(product);

                break;

            } catch (InputMismatchException imx) {
                System.out.println("--> Enter digits

                scanner.nextLine();
            } catch (Exception x) {
                System.out.println(x.getMessage());
            }
        }
    }
}

```

```

                                scanner.nextLine();
                                }
                                }
                                }

                                Customer updateCustomer = new
Customer(updatedpList);;
                                customers.updateTarget(updateID, updateCustomer,
2);

                                }

                                System.out.println("\n--> Data with ID: " + updateID
+ " is updated successfully.");
                                }
                                else
                                System.out.println("--> There is no data that matches
your ID.");

                                System.out.print("\nEnter any number to return: ");
                                choice = scanner.nextInt();
                                break;

//5. Delete Data
case 5:

                                System.out.println("\n-----\nDELETE DATA\n");
                                System.out.print("Enter customer ID to delete: ");
                                int deleteID = scanner.nextInt();

                                if (customers.isTheTargetOnTheList(deleteID) == true) {
                                        System.out.println("Are you sure to delete all data
for ID: " + deleteID + "?");
                                        System.out.print("Enter 1 to confirm or any number to
cancel: ");
                                        choice = scanner.nextInt();

                                        if (choice == 1) {
                                                customers.deleteTarget(deleteID);
                                                System.out.println("\n--> Data deleted
successfully.");
                                        }
                                        else
                                                break;
                                }
                                else
                                System.out.println("--> There is no data that matches
your ID.");

                                System.out.print("\nEnter any number to return: ");
                                choice = scanner.nextInt();
                                break;

//6. Exit
case 6:

                                System.out.println("\nAre you sure to exit program?");
                                System.out.print("Enter 1 to confirm, or any number to
cancel: ");
                                choice = scanner.nextInt();

                                if (choice == 1) {
                                        System.out.println("\nThank You!");
                                        System.exit(0); //terminates the program
                                }

                                break;

```

```
                default:
                    System.out.println("--> Enter digits from 1-6 only! Try
again.");
                    break;
            }
        }
        catch (InputMismatchException imx) {
            System.out.println("--> Enter digits only! Try again.");
            scanner.nextLine();
        }
        System.out.println("");
    }
}
```

SOURCE CODE (WITH GUI)

Attached below is the complete source code for the program **with GUI**, separated into four parts that represent the four Java classes used:

- *Product.java*
- *Customer.java*
- *Node.java*
- *CompanyABCManagementSystemGUI.java* (**Main class**)

--- Product.java

```
package customer;

public class Product {
    String Productsname;
    int Purchasequantity;
    String Purchasedate;

    public Product (String Productsname,int Purchasequantity, String
Purchasedate) {
        this.Productsname = Productsname;
        this.Purchasequantity = Purchasequantity;
        this.Purchasedate = Purchasedate;
    }

    public Product() {
    }
    public String getProductsename() {
        return Productsname;
    }
    public int getPurchasequantity() {
        return Purchasequantity;
    }

    public String getPurchasedate(){
        return Purchasedate;
    }
    public void setProdurtsename(String Produrtsname) {
        this.Productsname = Produrtsname;
    }

    public void setPurchasequantity(int Purchasequantity) {
        this.Purchasequantity = Purchasequantity;
    }
    public void setPurchasedate(String Purchasedate) {
        this.Purchasedate = Purchasedate;
    }
}
```


--- Customer.java

```

package customer;
import java.util.ArrayList;
public class Customer{
    int customerID;
    String name;
    long phoneNumber;
    private ArrayList<Product> pList;

    public Customer(int customerID, String name, long phoneNumber,
ArrayList<Product> pList) {
        this.customerID = customerID;
        this.name = name;
        this.phoneNumber = phoneNumber;
        this.pList = pList;
    }

    //constructor for updated customer personal info object
    public Customer(String name, int phoneNumber) {
        this.name = name;
        this.phoneNumber = phoneNumber;
    }

    //constructor for updated customer products info object
    public Customer(ArrayList<Product> pList) {
        this.pList = pList;
    }
    public int getCustomerID() {
        return customerID;
    }
    public String getCustomerName() {
        return name;
    }
    public ArrayList<Product> getPList() {
        return pList;
    }
    public long getCustomerphoneNumber(){
        return phoneNumber;
    }
    public void setCustomerName(String name) {
        this.name = name;
    }

    public void setCustomerphoneNumber(long phoneNumber) {
        this.phoneNumber = phoneNumber;
    }
}

```

--- Node.java

```

package customer;

class Node {
    private Customer info;
    private Node link;
}

```

```

public Node() {
    this.info = null;
    this.link = null;
}

public Node(Customer newInfo) {
    this.info = newInfo;
    this.link = null;
}

public void setInfo(Customer a) {
    this.info = a;
}

public void setLink(Node b) {
    this.link = b;
}

public Customer getInfo() {
    return this.info;
}

public Node getLink() {
    return this.link;
}
}

```

--- CompanyABCManagementSystemGUI.java

```

package customer;
import java.util.*;
import java.io.*;
import javax.swing.JOptionPane;
public class CompanyABCManagementSystemGUI extends javax.swing.JFrame {
    Node head ;
    int customerID = 1;
    int searchCount = 0;
    String Productsname = null;
    public CompanyABCManagementSystemGUI() {
        initComponents();
    }
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        productType = new javax.swing.ButtonGroup();
        jPanel1 = new javax.swing.JPanel();
        outputDisplay = new javax.swing.JButton();
        jPanel4 = new javax.swing.JPanel();
        jLabel22 = new javax.swing.JLabel();
        jLabel23 = new javax.swing.JLabel();
        jLabel24 = new javax.swing.JLabel();
        jLabel25 = new javax.swing.JLabel();
        jLabel26 = new javax.swing.JLabel();
        Name = new javax.swing.JTextField();
        Phone = new javax.swing.JTextField();
        Quantity = new javax.swing.JTextField();
        Date = new javax.swing.JTextField();
        jButton17 = new javax.swing.JButton();
        jButton18 = new javax.swing.JButton();
        jButton19 = new javax.swing.JButton();
    }
}

```

```

Search = new javax.swing.JTextField();
jButton20 = new javax.swing.JButton();
jButton21 = new javax.swing.JButton();
jLabel27 = new javax.swing.JLabel();
jLabel28 = new javax.swing.JLabel();
Type1 = new javax.swing.JRadioButton();
Type2 = new javax.swing.JRadioButton();
Type3 = new javax.swing.JRadioButton();
No = new javax.swing.JLabel();
jLabel1 = new javax.swing.JLabel();
jButton7 = new javax.swing.JButton();
jPanel2 = new javax.swing.JPanel();
jScrollPane2 = new javax.swing.JScrollPane();
output = new javax.swing.JTextArea();
jPanel3 = new javax.swing.JPanel();
jScrollPane1 = new javax.swing.JScrollPane();
output2 = new javax.swing.JTextArea();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jPanel1.setBackground(new java.awt.Color(204, 204, 204));

jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED), "Company ABC Management System", javax.swing.border.TitledBorder.CENTER, javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Segoe UI", 1, 12))); // NOI18N

outputDisplay.setText("Display");
outputDisplay.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        outputDisplayActionPerformed(evt);
    }
});

jPanel4.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 0, 0), 2, true));

jLabel22.setText("Customer Name :");
jLabel23.setText("Phone Number :");
jLabel24.setText("Product Purchase :");
jLabel25.setText(" Quantity :");
jLabel26.setText("Purchase Date :");

Name.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        customerNameActionPerformed(evt);
    }
});

jButton17.setText("Add Data");
jButton17.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

jButton18.setText("Update Data");
jButton18.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});

```

```

jButton19.setText("Delete Data");
jButton19.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton4ActionPerformed(evt);
    }
});

Search.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        searchActionPerformed(evt);
    }
});

jButton20.setText("Search");
jButton20.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

jButton21.setText("Reset");
jButton21.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton6ActionPerformed(evt);
    }
});

jLabel27.setText("DD/MM/YY");

jLabel28.setFont(new java.awt.Font("Segoe UI Symbol", 1, 14)); // NOI18N
jLabel28.setText("INFORMATION");

productType.add(Type1);
Type1.setText("Air Purifiers");
Type1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        Type1ActionPerformed(evt);
    }
});

productType.add(Type2);
Type2.setText("Water Purifiers");
Type2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        type2ActionPerformed(evt);
    }
});

productType.add(Type3);
Type3.setText("Air Conditioner");
Type3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        Type3ActionPerformed(evt);
    }
});

jLabel11.setText("Customer ID      : ");

jButton7.setText("EXIT");
jButton7.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton7ActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel4Layout = new
javax.swing.GroupLayout(jPanel4);
jPanel4.setLayout(jPanel4Layout);

```

```

jPanel4Layout.setHorizontalGroup(
jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup())
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel4Layout.createSequentialGroup())
                .addGap(110, 110, 110)
                .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                    .addComponent(jLabel22,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addComponent(jLabel23,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addComponent(jLabel24,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addComponent(jLabel25,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addComponent(jLabel26,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addComponent(jLabel11,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                        .addComponent(Phone,
javax.swing.GroupLayout.DEFAULT_SIZE, 173, Short.MAX_VALUE)
                        .addComponent(Quantity)
                        .addComponent(Date))
                    .addGroup(jPanel4Layout.createSequentialGroup())
                        .addComponent(Type1,
javax.swing.GroupLayout.PREFERRED_SIZE, 98,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(Type2))
                    .addComponent(Name,
javax.swing.GroupLayout.PREFERRED_SIZE, 173,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(No,
javax.swing.GroupLayout.PREFERRED_SIZE, 78,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(jPanel4Layout.createSequentialGroup())
                    .addContainerGap()
                    .addComponent(jButton7,
javax.swing.GroupLayout.PREFERRED_SIZE, 99,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(18, 18, 18)
                    .addComponent(jButton21,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(12, 12, 12)
                .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel4Layout.createSequentialGroup())
                        .addGap(21, 21, 21)
                        .addComponent(jButton17,
javax.swing.GroupLayout.PREFERRED_SIZE, 88,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlaceme
nt.RELATED)

        .addComponent(jButton18)
        .addGap(12, 12, 12)
        .addComponent(jButton19))
        .addComponent(Type3, javax.swing.GroupLayout.PREFERRED_SIZE,
117, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(0, 6, Short.MAX_VALUE))
        .addGroup(jPanel4Layout.createSequentialGroup())
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.LEADING)
        .addGroup(jPanel4Layout.createSequentialGroup())
        .addGap(277, 277, 277)
        .addComponent(jLabel27,
javax.swing.GroupLayout.PREFERRED_SIZE, 70,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(jPanel4Layout.createSequentialGroup())
        .addGap(255, 255, 255)
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.G
roupLayout.Alignment.LEADING)
        .addGroup(jPanel4Layout.createSequentialGroup())
        .addComponent(Search,
javax.swing.GroupLayout.PREFERRED_SIZE, 209,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.Componen
tPlacement.UNRELATED)
        .addComponent(jButton20,
javax.swing.GroupLayout.PREFERRED_SIZE, 88,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(jLabel28,
javax.swing.GroupLayout.PREFERRED_SIZE, 130,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
    jPanel4Layout.setVerticalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup())
    .addGap(12, 12, 12)
    .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.BASELINE)
    .addComponent(Search, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jButton20))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED)
    .addComponent(jLabel28)
    .addGap(7, 7, 7)
    .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.BASELINE)
    .addComponent(No)
    .addComponent(jLabel11))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNREL
ATED)
    .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.BASELINE)
    .addComponent(Name, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel22))
    .addGap(18, 18, 18)
    .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.BASELINE)
    .addComponent(Phone, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel23))
    .addGap(18, 18, 18)
    .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayo

```

```

ut.Alignment.BASELINE)
        .addComponent(jLabel124)
        .addComponent(Type1)
        .addComponent(Type2)
        .addComponent(Type3))
        .addGap(18, 18, 18)
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.
ut.Alignment.BASELINE)
        .addComponent(Quantity,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel125))
        .addGap(18, 18, 18)
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.
ut.Alignment.BASELINE)
        .addComponent(jLabel126)
        .addComponent(Date, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELAT
ED, 7, Short.MAX_VALUE)
        .addComponent(jLabel127, javax.swing.GroupLayout.PREFERRED_SIZE,
10, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.
ut.Alignment.BASELINE)
        .addComponent(jButton17)
        .addComponent(jButton18)
        .addComponent(jButton21)
        .addComponent(jButton19)
        .addComponent(jButton7))
        .addGap(15, 15, 15))
    );

    jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(null,
"Customers List", javax.swing.border.TitledBorder.CENTER,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Segoe UI",
1, 12))); // NOI18N

    output.setColumns(20);
    output.setRows(5);
    jScrollPane2.setViewportView(output);

    javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
    jPanel2.setLayout(jPanel2Layout);
    jPanel2Layout.setHorizontalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT_SIZE,
328, Short.MAX_VALUE)
            .addContainerGap())
    );
    jPanel2Layout.setVerticalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT_SIZE,
270, Short.MAX_VALUE)
            .addContainerGap())
    );

    jPanel3.setBorder(javax.swing.BorderFactory.createTitledBorder(null,
"Products List", javax.swing.border.TitledBorder.CENTER,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Segoe UI",

```

```

1, 12))); // NOI18N

        output2.setColumns(20);
        output2.setRows(5);
        jScrollPane1.setViewportViewView(output2);

        javax.swing.GroupLayout jPanel3Layout = new
javax.swing.GroupLayout(jPanel3);
        jPanel3.setLayout(jPanel3Layout);
        jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel3Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE,
325, Short.MAX_VALUE)
                .addContainerGap()
            )
            .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE,
325, Short.MAX_VALUE)
            )
        );
        jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel3Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE,
325, Short.MAX_VALUE)
            )
        );

        javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(16, 16, 16)
                        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                            .addGroup(jPanel1Layout.createSequentialGroup()
                                .addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                .addComponent(jPanel3,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addGap(9, 9, 9)
                                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                    .addComponent(jPanel4,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                    .addGroup(jPanel1Layout.createSequentialGroup()
                                        .addGap(317, 317, 317)
                                        .addComponent(outputDisplay,
javax.swing.GroupLayout.PREFERRED_SIZE, 110,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                                    .addGap(16, 16, 16)
                                )
                            .addGap(16, 16, 16)
                        )
                    )
                .addContainerGap()
            )
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE,
325, Short.MAX_VALUE)
            )
        );
        jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(19, 19, 19)
                        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
325, Short.MAX_VALUE)
                            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE,
325, Short.MAX_VALUE)
                        )
                    )
                .addContainerGap()
            )
        );

```



```

javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNREL
ATED)
        .addComponent(outputDisplay)
        .addGap(12, 12, 12)
        .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(11, 11, 11))
    );

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
Short.MAX_VALUE)
        )
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
Short.MAX_VALUE)
            )
        );

    pack();
} // </editor-fold>

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt)
{
    //exit
    System.exit(0);
}

private void type2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt)
{
    //reset
    No.setText("");
    Name.setText("");
    Phone.setText("");
    Quantity.setText("");
    Date.setText("");
    Search.setText("");
    output.setText("");
    output2.setText("");
    productType.clearSelection();
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
    //search
    output.setText("");
    output2.setText("");
    try{
        if (Search.getText().isEmpty() ){
            throw new Exception ("Please fill in customer ID!");
        }
        int target = Integer.parseInt(Search.getText());

```

```

        Search(target);
    } catch (IllegalArgumentException x){
        JOptionPane.showMessageDialog(null,"Please enter correct customer
ID!");
    } catch (Exception e){
        JOptionPane.showMessageDialog(null, e.getMessage());
    }
}

private void searchActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt)
{ //delete
    try{
        if (Search.getText().isEmpty() ){
            throw new Exception ("Please fill in customer ID in search
text filed!");
        }
        int target = Integer.parseInt(Search.getText());
        int respond = JOptionPane.showConfirmDialog(this,"Do you confirm to
delete the information?", "Confirmation", JOptionPane.YES_NO_OPTION,
JOptionPane.QUESTION_MESSAGE);
        if (respond == JOptionPane.YES_OPTION){
            deleteTarget(target);
            No.setText("");
            Name.setText("");
            Phone.setText("");
            Quantity.setText("");
            Date.setText("");
            Search.setText("");
            output.setText("");
            output2.setText("");
            productType.clearSelection();
        }
        else{
            JOptionPane.showMessageDialog(null,"Delete cancel!");
            No.setText("");
            Name.setText("");
            Phone.setText("");
            Quantity.setText("");
            Date.setText("");
            Search.setText("");
            output.setText("");
            output2.setText("");
            productType.clearSelection();
        }
    } catch (Exception e){
        JOptionPane.showMessageDialog(null, e.getMessage());
    }
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
{ //update
    ArrayList<Product> updatedpList = new ArrayList<>();
    try{
        if (Search.getText().isEmpty() ){
            throw new Exception ("Please fill in customer ID in search
text filed!");
        }
        int respond = JOptionPane.showConfirmDialog(this,"Do you confirm to
update new information?", "Confirmation", JOptionPane.YES_NO_OPTION,
JOptionPane.QUESTION_MESSAGE);
        if (respond == JOptionPane.YES_OPTION){
            int target = Integer.parseInt(Search.getText());
            int customerID = Integer.parseInt(No.getText());
            String name = Name.getText();

```

```

        long phoneNumber = Long.parseLong(Phone.getText());
        if (Type1.isSelected()){
            Productsname = "Air Purifiers";
        }
        else if (Type2.isSelected()){
            Productsname = "Water Purifiers";
        }
        else if (Type3.isSelected()){
            Productsname = "Air Conditioner";
        }
        int Purchasequantity = Integer.parseInt(Quantity.getText());
        String Purchasedate = Date.getText();
        if (phoneNumber < 0 || Purchasequantity < 0 ){
            throw new Exception ("Failed to Add.Please enter the correct
information!");
        }
        Product product = new Product(Productsname, Purchasequantity,
Purchasedate);
        updatedpList.add(product);
        Customer newInfo = new Customer(customerID,name,phoneNumber,
updatedpList);
        updateTarget(target,newInfo);
        JOptionPane.showMessageDialog(null,"Sucessful Update!");
        No.setText("");
        Search.setText("");
        Name.setText("");
        Phone.setText("");
        Quantity.setText("");
        Date.setText("");
        Search.setText("");
        output.setText("");
        output2.setText("");
        productType.clearSelection();
    }
    else {
        JOptionPane.showMessageDialog(null,"Update Cancel!");
        No.setText("");
        output.setText("");
        output2.setText("");
        Search.setText("");
        Name.setText("");
        Phone.setText("");
        Quantity.setText("");
        Date.setText("");
        Search.setText("");
        productType.clearSelection();
    }
} catch (IllegalArgumentException x){
    JOptionPane.showMessageDialog(null,"Some information entered is
incorrect or empty. Please enter again!");
} catch (Exception e){
    JOptionPane.showMessageDialog(null,e.getMessage());
}
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
    //add
    ArrayList<Product> pList = new ArrayList<>();
    try{
        String name = Name.getText();
        long phoneNumber = Long.parseLong(Phone.getText());
        if (Type1.isSelected()){
            Productsname = "Air Purifiers";
        }
        else if (Type2.isSelected()){
            Productsname = "Water Purifiers";
        }
        else if (Type3.isSelected()){

```

```

        Productsname = "Water Purifiers";
    }

    int Purchasequantity = Integer.parseInt(Quantity.getText());
    String Purchasedate = Date.getText();
    if (phoneNumber < 0 || Purchasequantity < 0 ){
        throw new Exception ("Failed to Add.Please enter the correct
information!");
    }
    Product product = new Product(Productsname, Purchasequantity,
Purchasedate);
    pList.add(product);
    Customer data = new Customer(customerID,name,phoneNumber,pList);
    customerID++;
    addData(data);
    JOptionPane.showMessageDialog(null,"Sucessful Add!");
    No.setText("");
    output.setText("");
    output2.setText("");
    Search.setText("");
    Name.setText("");
    Phone.setText("");
    Quantity.setText("");
    Date.setText("");
    productType.clearSelection();
} catch (IllegalArgumentException x){
    JOptionPane.showMessageDialog(null,"Some information entered is
incorrect or empty. Please enter again!");
} catch (Exception e){
    JOptionPane.showInternalMessageDialog(null,e.getMessage());
}

}

private void customerNameActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void outputDisplayActionPerformed(java.awt.event.ActionEvent evt)
{ //display
    output.setText("");
    output2.setText("");
    Node current = head;
    int index = 1;
    output.setText("");
    if(current == null)
        JOptionPane.showMessageDialog(null,"There is nothing in the list.");
    else
        while (current != null){
            output.append("-----Index" + index + "-----\n");
            output.append("Customer ID  :" + current.getInfo().getCustomerID()+
"\n" );
            output.append("Customer Name :" +
current.getInfo().getCustomerName()+ "\n");
            output.append("Customer PhoneNumber  :" +
String.valueOf(current.getInfo().getCustomerphoneNumber())+ "\n");

            ArrayList<Product> pList = current.getInfo().getPList();
            for (Product product : pList) {
                output.append("Product purchase  :" + product.getProductsname()+
"\n");
                output2.append("-----Index" + index + "-----\n");
                output2.append("Customer ID  :" + current.getInfo().getCustomerID()+
"\n" );
                output2.append("Product purchase  :" + product.getProductsname()+
"\n");
                output2.append("Quantity  :" +
Integer.toString(product.getPurchasequantity())+ "\n");
                output2.append("Purchase Date  :" + product.getPurchasedate()+ "\n");
            }
        }
    }
}

```

```

        }
        index++;
        current = current.getLink();
    }
}

private void Type1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void Type3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the
default look and feel.
     * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(CompanyABCManagementSystemGUI.class.getName())
.log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(CompanyABCManagementSystemGUI.class.getName())
.log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(CompanyABCManagementSystemGUI.class.getName())
.log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(CompanyABCManagementSystemGUI.class.getName())
.log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new CompanyABCManagementSystemGUI().setVisible(true);
        }
    });
}

public void addData(Customer a) { //add method
    Node newNode = new Node(a);
    if (head == null)
        head = newNode;
    else {
        Node current = head;
        while (current.getLink() != null)

```

```

        current = current.getLink();
        current.setLink(newNode);
    }
    searchCount++;
}

public void deleteTarget(int target) { //delete method
    if (head == null){
        JOptionPane.showMessageDialog(null,"There is nothing in
the list.");
    }
    else if (head.getInfo().getCustomerID()== target){
        head = head.getLink();
        JOptionPane.showMessageDialog(null,"Delete successful");
    }
    else {
        Node before = null;
        Node current = head;

        while ((current.getInfo().getCustomerID() != target) &&
(current.getLink() != null)) {
            before = current;
            current = current.getLink();
        }
        if (current.getInfo().getCustomerID() == target) {
            before.setLink(current.getLink());
            output.setText("Delete successful");
            searchCount--;
        } else
            JOptionPane.showMessageDialog(null,"There is nothing matches the item
that you wish to delete");
    }
}

public void Search(int target) { //search method
    Node current = head;
    if (current == null){
        JOptionPane.showMessageDialog(null, "There is nothing in
the list!");
    }
    else{
        while (current.getInfo().getCustomerID() != target &&
current.getLink() != null) {
            current = current.getLink();
        }
        if (current.getInfo().getCustomerID() == target){
No.setText(String.valueOf(current.getInfo().getCustomerID()));
Name.setText(current.getInfo().getCustomerName());
Phone.setText(String.valueOf(current.getInfo().getCustomerphoneNumber()));

        ArrayList<Product> pList =
current.getInfo().getPList();
        for (Product product : pList) {
            if (product.getProductsename().equals("Air
Purifiers")){
                Type1.setSelected(true);
            }
            else if (product.getProductsename().equals("Water
Purifiers")){
                Type2.setSelected(true);
            }
            else {
                Type3.setSelected(true);
            }
        }
        Quantity.setText(Integer.toString(product.getPurchasequantity()));
        Date.setText(product.getPurchasedate());
    }
}

```

```

        }
    }
    else{
        JOptionPane.showMessageDialog(null, "There is noting matches your
search");
    }
}

    public void updateTarget(int target, Customer newInfo) { //update method
    if (head.getInfo().getCustomerID() == target){
        head.getInfo().setCustomerName(newInfo.getCustomerName());

        head.getInfo().setCustomerphoneNumber(newInfo.getCustomerphoneNumber());
        ArrayList<Product> pList = head.getInfo().getPList();
        pList.clear();

        ArrayList<Product> updatedpList = newInfo.getPList();
        pList.addAll(updatedpList);
    }
    else {
        Node current = head;
        while ((current.getInfo().getCustomerID() != target) &&
(current.getLink() != null)) {
            current = current.getLink();
        }
        if (current.getInfo().getCustomerID() == target) {
            current.getInfo().setCustomerName(newInfo.getCustomerName());

            current.getInfo().setCustomerphoneNumber(newInfo.getCustomerphoneNumber());{
                ArrayList<Product> pList = current.getInfo().getPList();
                pList.clear(); // Clear the existing products

                ArrayList<Product> updatedpList = newInfo.getPList();
                pList.addAll(updatedpList); //add updated products list
            }
        }
        else
            JOptionPane.showMessageDialog(null,"Error. There is nothing matches
the item that you wish to update");
    }
}

// Variables declaration - do not modify
private javax.swing.JTextField Date;
private javax.swing.JTextField Name;
private javax.swing.JLabel No;
private javax.swing.JTextField Phone;
private javax.swing.JTextField Quantity;
private javax.swing.JTextField Search;
private javax.swing.JRadioButton Type1;
private javax.swing.JRadioButton Type2;
private javax.swing.JRadioButton Type3;
private javax.swing.JButton jButton17;
private javax.swing.JButton jButton18;
private javax.swing.JButton jButton19;
private javax.swing.JButton jButton20;
private javax.swing.JButton jButton21;
private javax.swing.JButton jButton7;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel22;
private javax.swing.JLabel jLabel23;
private javax.swing.JLabel jLabel24;
private javax.swing.JLabel jLabel25;
private javax.swing.JLabel jLabel26;
private javax.swing.JLabel jLabel27;
private javax.swing.JLabel jLabel28;

```

```
private javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel2;  
private javax.swing.JPanel jPanel3;  
private javax.swing.JPanel jPanel4;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JScrollPane jScrollPane2;  
private javax.swing.JTextArea output;  
private javax.swing.JTextArea output2;  
private javax.swing.JButton outputDisplay;  
private javax.swing.ButtonGroup productType;  
// End of variables declaration  
}
```


SAMPLE OUTPUT (WITHOUT GUI)

Below is the sample output of the console-based program for Company ABC being executed in Apache NetBeans IDE 17, with **user input in orange**:

```
run:
*****
Welcome To Company ABC

1. Display Data
2. Add Data
3. Search Data
4. Update Data
5. Delete Data
6. Exit

Enter your choice: 1

-----
DISPLAY LIST

--> The list is empty, add new data!

Enter any number to return: 1

*****
Welcome To Company ABC

1. Display Data
2. Add Data
3. Search Data
4. Update Data
5. Delete Data
6. Exit

Enter your choice: 2

-----
ADD DATA

Enter customer name: Muhammad Zuhair
Enter customer phone number: iajdsoajd
--> Phone number can only have digits! Try again.
Enter customer phone number: 0123456789
Enter number of products purchased: 0
--> Invalid digit! Try again.
Enter number of products purchased: 2

Product 1
-- 1. Air purifier
-- 2. Water purifier
-- 3. Air conditioner
Select product type for Product 1: 9
--> Invalid digit! Try again.

Product 1
-- 1. Air purifier
-- 2. Water purifier
-- 3. Air conditioner
Select product type for Product 1: 1
Enter purchase date: 3 July 2023
Enter quantity: 0
```

```
--> Invalid quantity! Try again.
Enter quantity: 3

Product 2
-- 1. Air purifier
-- 2. Water purifier
-- 3. Air conditioner
Select product type for Product 2: 3
Enter purchase date: 10 July 2023
Enter quantity: 1

--> Customer added successfully.

Enter any number to return: 9

*****
Welcome To Company ABC

1. Display Data
2. Add Data
3. Search Data
4. Update Data
5. Delete Data
6. Exit

Enter your choice: 1

-----
DISPLAY LIST

|ID.|CUSTOMER NAME      |PHONE NUMBER |      PRODUCT NAME|  QUANTITY|      PURCHASE DATE|
|1 |Muhammad Zuhair   |0123456789  |      Air purifier|        3|      3 July 2023|
|  |                  |            |      Air conditioner|        1|     10 July 2023|

Enter any number to return: 4

*****
Welcome To Company ABC

1. Display Data
2. Add Data
3. Search Data
4. Update Data
5. Delete Data
6. Exit

Enter your choice: 2

-----
ADD DATA

Enter customer name: Edwin Limmmmmmmmm
Enter customer phone number: 013
Enter number of products purchased: 2

Product 1
-- 1. Air purifier
-- 2. Water purifier
-- 3. Air conditioner
Select product type for Product 1: 2
Enter purchase date: 1 August 2023
Enter quantity: 5

Product 2
-- 1. Air purifier
-- 2. Water purifier
-- 3. Air conditioner
Select product type for Product 2: 1
Enter purchase date: 1 August 2023
Enter quantity: 1
```

--> Customer added successfully.

Enter any number to return: 3

Welcome To Company ABC

1. Display Data
2. Add Data
3. Search Data
4. Update Data
5. Delete Data
6. Exit

Enter your choice: 1

DISPLAY LIST

ID.	CUSTOMER NAME	PHONE NUMBER	PRODUCT NAME	QUANTITY	PURCHASE DATE
1	Muhammad Zuhair	0123456789	Air purifier	3	3 July 2023
			Air conditioner	1	10 July 2023
2	Edwin Limmmmmmm	013	Water purifier	5	1 August 2023
			Air purifier	1	1 August 2023

Enter any number to return: 0

Welcome To Company ABC

1. Display Data
2. Add Data
3. Search Data
4. Update Data
5. Delete Data
6. Exit

Enter your choice: 3

SEARCH DATA

Please enter Customer ID for Search : 3

--> There is no data that matches your ID.

Enter any number to return: 3

Welcome To Company ABC

1. Display Data
2. Add Data
3. Search Data
4. Update Data
5. Delete Data
6. Exit

Enter your choice: 3

SEARCH DATA

Please enter Customer ID for Search : 2

ID.	CUSTOMER NAME	PHONE NUMBER	PRODUCT NAME	QUANTITY	PURCHASE DATE
2	Edwin Limmmmmmm	013	Water purifier	5	1 August 2023
			Air purifier	1	1 August 2023

Enter any number to return: 0

```

Welcome To Company ABC

1. Display Data
2. Add Data
3. Search Data
4. Update Data
5. Delete Data
6. Exit

Enter your choice: 4

-----
UPDATE DATA

Enter customer ID to update: 2

Select info to update for customer ID 2
-- 1. Update Personal Info (Name, Phone Number)
-- 2. Update Products Info
Enter number 1 or 2 to update respective info, or any other number to cancel: 1
Enter customer name: Edwin Lim
Enter customer phone number: 013456789www
--> Phone number can only have digits! Try again.
Enter customer phone number: 0134567890

--> Data with ID: 2 is updated successfully.

Enter any number to return: 1

*****
Welcome To Company ABC

1. Display Data
2. Add Data
3. Search Data
4. Update Data
5. Delete Data
6. Exit

Enter your choice: 4

-----
UPDATE DATA

Enter customer ID to update: 2

Select info to update for customer ID 2
-- 1. Update Personal Info (Name, Phone Number)
-- 2. Update Products Info
Enter number 1 or 2 to update respective info, or any other number to cancel: 2
Enter number of products purchased: 1

Product 1
-- 1. Air purifier
-- 2. Water purifier
-- 3. Air conditioner
Select product type for Product 1: 2
Enter purchase date: 1 August 2023
Enter quantity: 5

--> Data with ID: 2 is updated successfully.

Enter any number to return: 1

*****
Welcome To Company ABC

1. Display Data
2. Add Data
3. Search Data
4. Update Data
5. Delete Data

```

6. Exit

Enter your choice: 3

SEARCH DATA

Please enter Customer ID for Search : 2

ID.	CUSTOMER NAME	PHONE NUMBER	PRODUCT NAME	QUANTITY	PURCHASE DATE
2	Edwin Lim	0134567890	Water purifier	5	1 August 2023

Enter any number to return: 0

Welcome To Company ABC

1. Display Data
2. Add Data
3. Search Data
4. Update Data
5. Delete Data
6. Exit

Enter your choice: 1

DISPLAY LIST

ID.	CUSTOMER NAME	PHONE NUMBER	PRODUCT NAME	QUANTITY	PURCHASE DATE
1	Muhammad Zuhair	0123456789	Air purifier	3	3 July 2023
			Air conditioner	1	10 July 2023
2	Edwin Lim	0134567890	Water purifier	5	1 August 2023

Enter any number to return: 1

Welcome To Company ABC

1. Display Data
2. Add Data
3. Search Data
4. Update Data
5. Delete Data
6. Exit

Enter your choice: 5

DELETE DATA

Enter customer ID to delete: 2
Are you sure to delete all data for ID: 2?
Enter 1 to confirm or any number to cancel: 0

Welcome To Company ABC

1. Display Data
2. Add Data
3. Search Data
4. Update Data
5. Delete Data
6. Exit

Enter your choice: 5

DELETE DATA

Enter customer ID to delete: 1

```

Are you sure to delete all data for ID: 1?
Enter 1 to confirm or any number to cancel: 1

--> Data deleted successfully.

Enter any number to return: 1

*****
Welcome To Company ABC

1. Display Data
2. Add Data
3. Search Data
4. Update Data
5. Delete Data
6. Exit

Enter your choice: 1

-----
DISPLAY LIST

|ID.|CUSTOMER NAME      |PHONE NUMBER |      PRODUCT NAME|  QUANTITY|      PURCHASE DATE|
|2 |Edwin Lim          |0134567890  |      Water purifier|         5|      1 August 2023|

Enter any number to return: 0

*****
Welcome To Company ABC

1. Display Data
2. Add Data
3. Search Data
4. Update Data
5. Delete Data
6. Exit

Enter your choice: 3

-----
SEARCH DATA

Please enter Customer ID for Search : 1
--> There is no data that matches your ID.

Enter any number to return: 1

*****
Welcome To Company ABC

1. Display Data
2. Add Data
3. Search Data
4. Update Data
5. Delete Data
6. Exit

Enter your choice: 2

-----
ADD DATA

Enter customer name: Shaismeeta Kumar
Enter customer phone number: 0198765432
Enter number of products purchased: 3

Product 1
-- 1. Air purifier
-- 2. Water purifier
-- 3. Air conditioner
Select product type for Product 1: 1

```

```

Enter purchase date: 25 February 2023
Enter quantity: 3

Product 2
-- 1. Air purifier
-- 2. Water purifier
-- 3. Air conditioner
Select product type for Product 2: 2
Enter purchase date: 9 April 2023
Enter quantity: 2

Product 3
-- 1. Air purifier
-- 2. Water purifier
-- 3. Air conditioner
Select product type for Product 3: 3
Enter purchase date: 30 June 2023
Enter quantity: 5

--> Customer added successfully.

Enter any number to return: 1

*****
Welcome To Company ABC

1. Display Data
2. Add Data
3. Search Data
4. Update Data
5. Delete Data
6. Exit

Enter your choice: 1

-----
DISPLAY LIST

|ID.|CUSTOMER NAME      |PHONE NUMBER |      PRODUCT NAME|  QUANTITY|      PURCHASE DATE|
|2 |Edwin Lim          |0134567890  |      Water purifier|         5|      1 August 2023|
|3 |Shaismeeta Kumar   |0198765432  |      Air purifier  |         3| 25 February 2023|
| |                   |             |      Water purifier|         2|      9 April 2023|
| |                   |             |      Air conditioner|         5|      30 June 2023|

Enter any number to return: 0

*****
Welcome To Company ABC

1. Display Data
2. Add Data
3. Search Data
4. Update Data
5. Delete Data
6. Exit

Enter your choice: 7
--> Enter digits from 1-6 only! Try again.

*****
Welcome To Company ABC

1. Display Data
2. Add Data
3. Search Data
4. Update Data
5. Delete Data
6. Exit

Enter your choice: jjj
--> Enter digits only! Try again.

```

```
*****
Welcome To Company ABC

1. Display Data
2. Add Data
3. Search Data
4. Update Data
5. Delete Data
6. Exit

Enter your choice: 6

Are you sure to exit program?
Enter 1 to confirm, or any number to cancel: 0

*****
Welcome To Company ABC

1. Display Data
2. Add Data
3. Search Data
4. Update Data
5. Delete Data
6. Exit

Enter your choice: 6

Are you sure to exit program?
Enter 1 to confirm, or any number to cancel: 1

Thank You!
BUILD SUCCESSFUL (total time: 6 minutes 37 seconds)
```


SAMPLE OUTPUT (WITH GUI)

1. Below is the main page of the system. The user may search, add, update, delete, or display data.

Company ABC Management System

Customers List

Products List

Display

Search

INFORMATION

Customer ID :

Customer Name :

Phone Number :

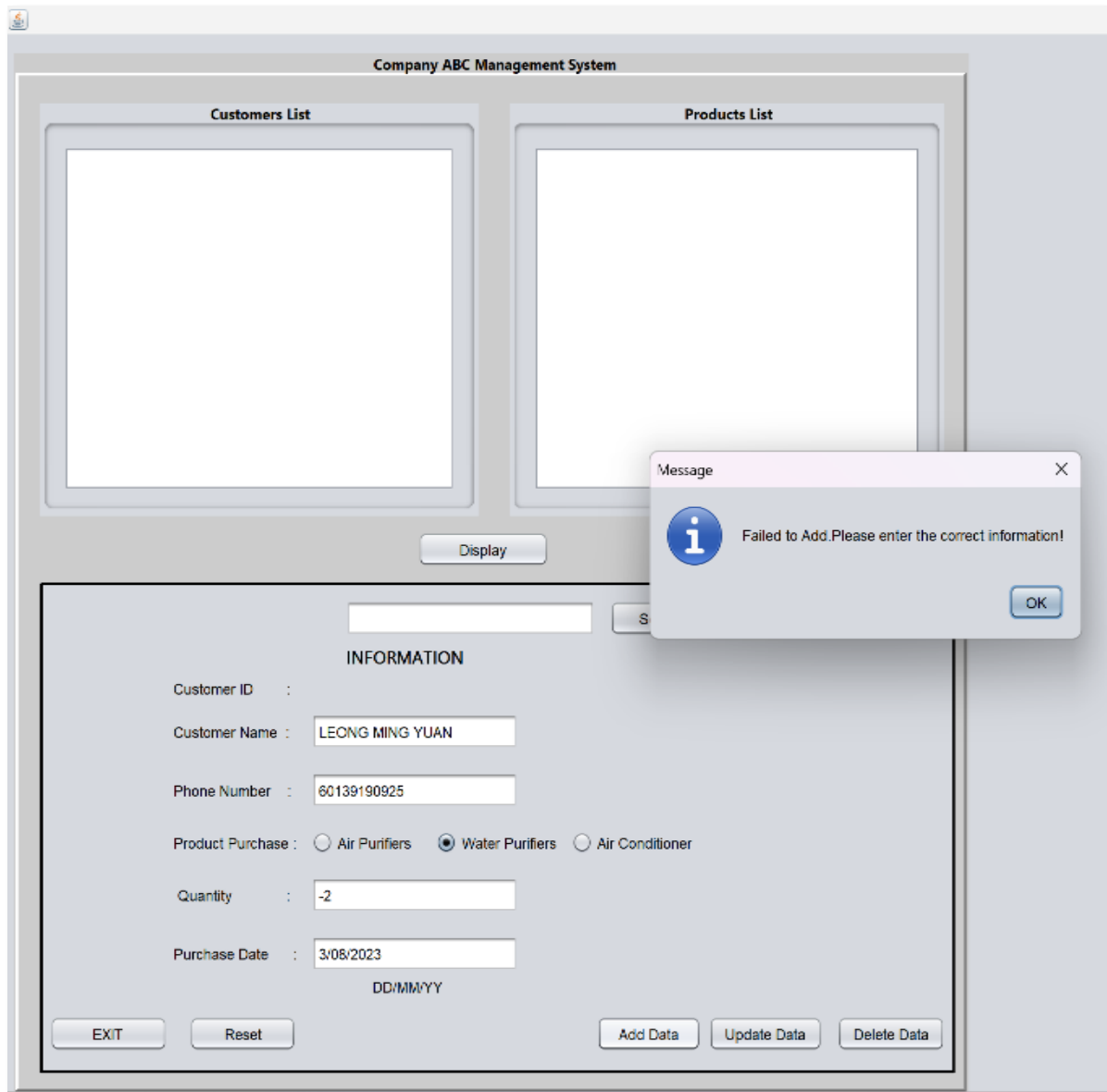
Product Purchase : ☐ Air Purifiers ☐ Water Purifiers ☐ Air Conditioner

Quantity :

Purchase Date : DD/MM/YY


EXIT Reset Add Data Update Data Delete Data

2. To add data, user needs to key in the information required and click the 'Add Data' button. The system will notify user when user enters an invalid input, or if there is no data being added yet. Customer ID is automatically generated, hence user cannot enter or change the customer ID.

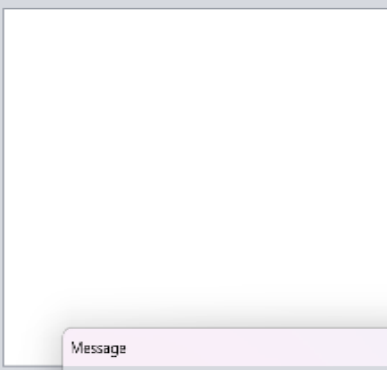


Company ABC Management System

Customers List



Products List



Display

INFORMATION

Customer ID :

Customer Name :

Phone Number :

Product Purchase : ☐ Air Purifiers ☒ Water Purifiers ☐ Air Conditioner


Quantity :

Purchase Date :

DD/MM/YY

EXIT Reset Add Data Update Data Delete Data

Message

 Some information entered is incorrect or empty. Please enter again!

OK

The screenshot displays the 'Company ABC Management System' window. It features two main sections: 'Customers List' and 'Products List', both of which are currently empty. Below these sections is a 'Display' button. The bottom section, titled 'INFORMATION', contains several input fields and buttons. A 'Search' button is located above a search input field. The 'INFORMATION' section includes fields for 'Customer ID', 'Customer Name' (filled with 'LEONG MING YUAN'), 'Phone Number' (filled with '60139190925'), 'Product Purchase' (with radio buttons for 'Air Purifiers', 'Water Purifiers' (selected), and 'Air Conditioner'), 'Quantity' (filled with '2'), and 'Purchase Date' (filled with '3/08/2023'). Below the date field is a label 'DD/MM/YY'. At the bottom of the 'INFORMATION' section are buttons for 'EXIT', 'Reset', 'Add Data', 'Update Data', and 'Delete Data'. A 'Message' dialog box is overlaid on the right side of the window, displaying an information icon, the text 'Sucessful Add!', and an 'OK' button.

Company ABC Management System

Customers List

Products List

Display

Search

INFORMATION

Customer ID :

Customer Name : LEONG MING YUAN

Phone Number : 60139190925

Product Purchase : ☐ Air Purifiers ☒ Water Purifiers ☐ Air Conditioner

Quantity : 2

Purchase Date : 3/08/2023

DD/MM/YY

EXIT Reset Add Data Update Data Delete Data

Message

Successfull Add!

OK

3. To search data, user needs to key in the ID of the customer they wish to search for and click the 'Search' button. The system will notify user when user enters an invalid customer ID or if the ID does not exist in the list. The system will show the information of the customer if the customer ID is found.

The screenshot displays the 'Company ABC Management System' window. It features two main panels: 'Customers List' and 'Products List', both of which are currently empty. Below these panels is a 'Display' button. At the bottom of the window is a form titled 'INFORMATION'. This form includes a search bar with the text 'MING' and a 'Search' button. Below the search bar, there are several input fields: 'Customer ID', 'Customer Name', 'Phone Number', 'Quantity', and 'Purchase Date'. The 'Purchase Date' field is accompanied by a date format indicator 'DD/MM/YY'. There are also three radio buttons for 'Product Purchase' with options 'Air Purifiers', 'Water Purifiers', and 'Air Conditioner'. At the bottom of the form are four buttons: 'EXIT', 'Reset', 'Add Data', and 'Update Data'. A 'Delete Data' button is also present. A 'Message' dialog box is overlaid on the right side of the window, displaying an information icon and the text 'Please enter correct customer ID!' with an 'OK' button.

Company ABC Management System

Customers List

Products List

Display

3 Search

Message

There is noting matches your search

OK

INFORMATION

Customer ID :

Customer Name :

Phone Number :

Product Purchase : ☐ Air Purifiers ☐ Water Purifiers ☐ Air Conditioner

Quantity :

Purchase Date :

DD/MM/YY

EXIT Reset Add Data Update Data Delete Data

Company ABC Management System

Customers List

Products List

Display

1 Search

INFORMATION

Customer ID : 1

Customer Name : LEONG MING YUAN

Phone Number : 60139190925

Product Purchase : ☐ Air Purifiers ☒ Water Purifiers ☐ Air Conditioner

Quantity : 2

Purchase Date : 3/08/2023

DD/MM/YY

EXIT Reset Add Data Update Data Delete Data

4. To display data, user needs to click the 'Display' button. The system will display all the data that has been added by the user.

The screenshot displays the 'Company ABC Management System' interface. It features two main data display panels at the top: 'Customers List' and 'Products List'. Both panels show a single record with the following details:

- Index1-----
- Customer ID :1
- Customer Name :LEONG MING YUAN
- Customer PhoneNumber :60139190925
- Product purchase :Water Purifiers

Below these panels is a 'Display' button. At the bottom of the interface is an 'INFORMATION' section containing input fields for Customer ID, Customer Name, Phone Number, Product Purchase (with radio buttons for Air Purifiers, Water Purifiers, and Air Conditioner), Quantity, and Purchase Date (with a DD/MM/YY format). This section includes an 'EXIT' button, a 'Reset' button, and three buttons for data management: 'Add Data', 'Update Data', and 'Delete Data'. A search bar with a 'Search' button is located at the top right of the information section.

5. To update data, user needs to search for the ID of the customer they wish to update in order to display their information. Users may then update the respective customer information and click the 'Update data' button. The system will show a confirmation message to the user once update is successful. The system will also notify users when they enter an invalid input or if they left an input field blank.

The screenshot displays the 'Company ABC Management System' window. It features two main panels: 'Customers List' and 'Products List', both of which are currently empty. Below these panels is a 'Display' button. At the bottom of the window is a form titled 'INFORMATION'. This form includes a search bar with the number '1' entered, a 'Search' button, and several input fields: 'Customer ID' (value: 1), 'Customer Name' (value: LEONG MING YUAN), 'Phone Number' (value: 60139190925), 'Product Purchase' (radio buttons for 'Air Purifiers' (selected), 'Water Purifiers', and 'Air Conditioner'), 'Quantity' (value: 3), and 'Purchase Date' (value: 3/08/2023). A date format indicator 'DD/MM/YY' is shown below the date field. At the bottom of the form are buttons for 'EXIT', 'Reset', 'Add Data', 'Update Data', and 'Delete Data'. A 'Confirmation' dialog box is overlaid on the right side of the window, asking 'Do you confirm to update new information?' with 'Yes' and 'No' buttons.

Company ABC Management System

Customers List

Products List

Display

1 Search

INFORMATION

Customer ID : 1

Customer Name : LEONG MING YUAN

Phone Number : 60139190925

Product Purchase : ☒ Air Purifiers ☐ Water Purifiers ☐ Air Conditioner

Quantity : 3

Purchase Date : 3/08/2023
DD/MM/YY

EXIT Reset Add Data Update Data Delete Data

Message

Update Cancel!

OK

Company ABC Management System

Customers List

Products List

Display

1 Search

INFORMATION

Customer ID : 1

Customer Name : LEONG MING YUAN

Phone Number : 60139190925

Product Purchase : ☒ Air Purifiers ☐ Water Purifiers ☐ Air Conditioner

Quantity : 3

Purchase Date : 3/08/2023
DD/MM/YY

EXIT Reset Add Data Update Data Delete Data

Message X

i Successful Update!

OK

Company ABC Management System

Customers List

-----Index1-----
Customer ID :1
Customer Name :LEONG MING YUAN
Customer PhoneNumber :60139190925
Product purchase :Air Purifiers

Products List

-----Index1-----
Customer ID :1
Product purchase :Air Purifiers
Quantity :3
Purchase Date :3/08/2023

Display

Search

INFORMATION

Customer ID :

Customer Name :

Phone Number :

Product Purchase : ☐ Air Purifiers ☐ Water Purifiers ☐ Air Conditioner

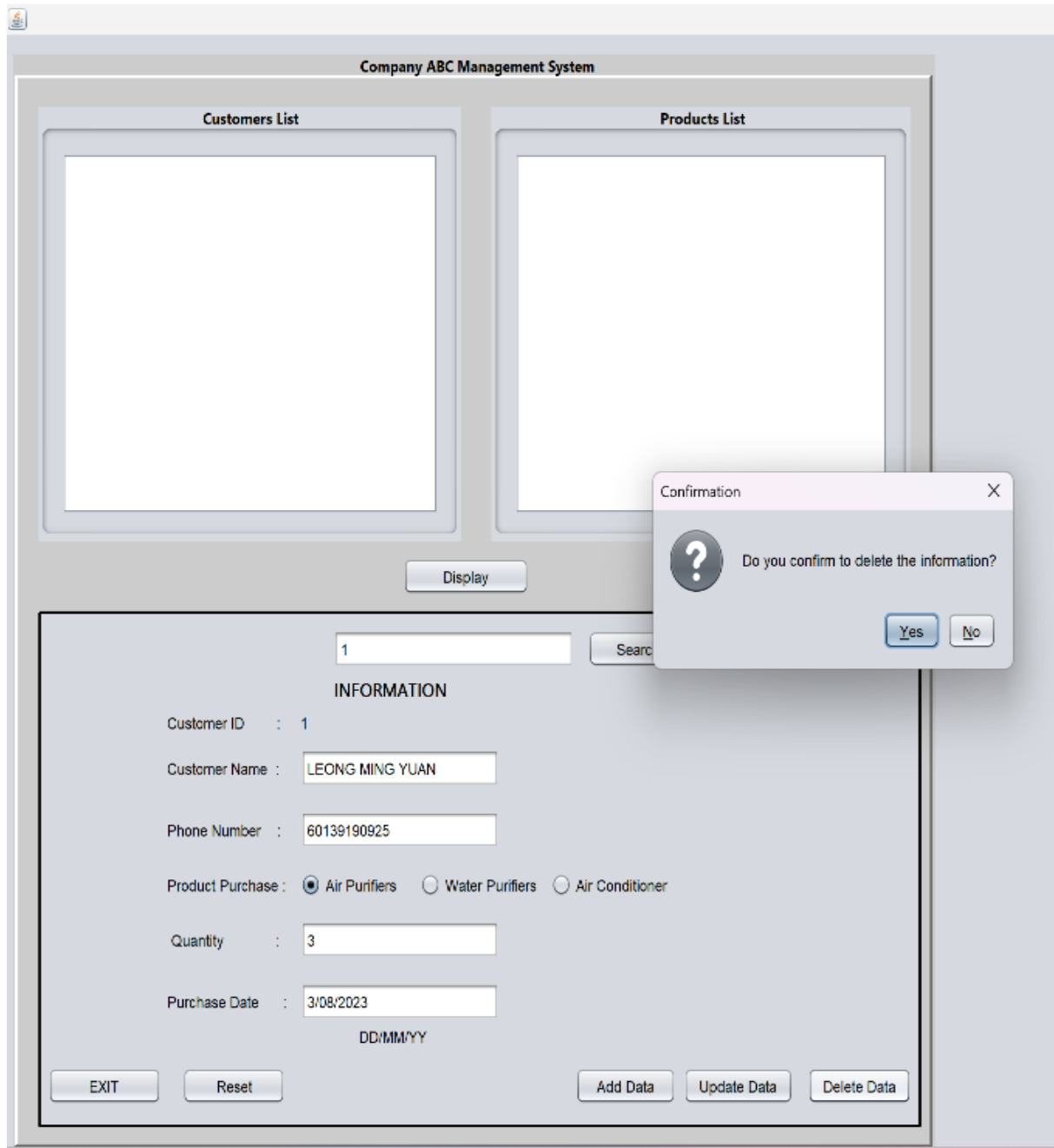
Quantity :

Purchase Date :

DD/MM/YY

EXIT Reset Add Data Update Data Delete Data

6. To delete data, user needs to search for the ID of the customer they wish to delete in order to display their information, then user is required to click the 'Delete data' button. The system will show a confirmation message to the user once deletion is successful.



The screenshot displays the 'Company ABC Management System' window. It features two main panels: 'Customers List' and 'Products List', both of which are currently empty. Below these panels is a 'Display' button. At the bottom of the window is an 'INFORMATION' section containing several input fields and buttons. A 'Message' dialog box is overlaid on the right side of the window, displaying an information icon and the text 'Delete cancel!' with an 'OK' button.

Company ABC Management System

Customers List

Products List

Display

INFORMATION

Customer ID : 1

Customer Name : LEONG MING YUAN

Phone Number : 60139190925

Product Purchase : ☒ Air Purifiers ☐ Water Purifiers ☐ Air Conditioner

Quantity : 3

Purchase Date : 3/08/2023

DD/MM/YY

Message

Delete cancel!

OK

Search

EXIT **Reset** **Add Data** **Update Data** **Delete Data**

Company ABC Management System

Customers List

Products List

Display

1 Search

INFORMATION

Customer ID :

Customer Name :

Phone Number :

Product Purchase : ☐ Air Purifiers ☐ Water Purifiers ☐ Air Conditioner

Quantity :

Purchase Date :

DD/MM/YY

EXIT Reset Add Data Update Data Delete Data

Message X

Delete successful

OK

The screenshot displays the 'Company ABC Management System' window. It features two main panels: 'Customers List' and 'Products List', both of which are currently empty. Below these panels is a 'Display' button. At the bottom of the window is an 'INFORMATION' section containing several input fields: a search bar with a 'Search' button, 'Customer ID', 'Customer Name', 'Phone Number', 'Product Purchase' (with radio buttons for 'Air Purifiers', 'Water Purifiers', and 'Air Conditioner'), 'Quantity', and 'Purchase Date' (with a 'DD/MM/YY' format hint). At the very bottom are buttons for 'EXIT', 'Reset', 'Add Data', 'Update Data', and 'Delete Data'. A 'Message' dialog box is overlaid on the right side of the window, displaying an information icon and the text 'There is nothing in the list.' with an 'OK' button.

PRESENTATION VIDEO LINK:

<https://youtu.be/PvAZJ7cqI5w>