Assignment Letter

Student information :

Name	: Christopher Alexander Tjiandra
Binusian ID	: 2502019230
Course Name	: Object Oriented Programming (COMP6699001)
Class	: L2AC
Name of Lecturer(s)	: Jude Joseph Lamug Martinez
Major	: Computer Science
Title of Assignment	: Mini Cafe (Warkop) App
Type of Assignment	: Final Project

Submission Pattern :

Due Date	: June, 10th 2022
Submission Date	: June, 5th 2022

The assignment should meet the below requirements.

- Project specification
- Solution Design
- Evidence of working program
- Resources

Plagiarism/Cheating

Binus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

Declaration of Originality

By signing this assignment, I understand, accept and consent to Binus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

The undersigned below,

(Christopher Alexander Tjiandra)

“Mini Cafe App (Warkop)”

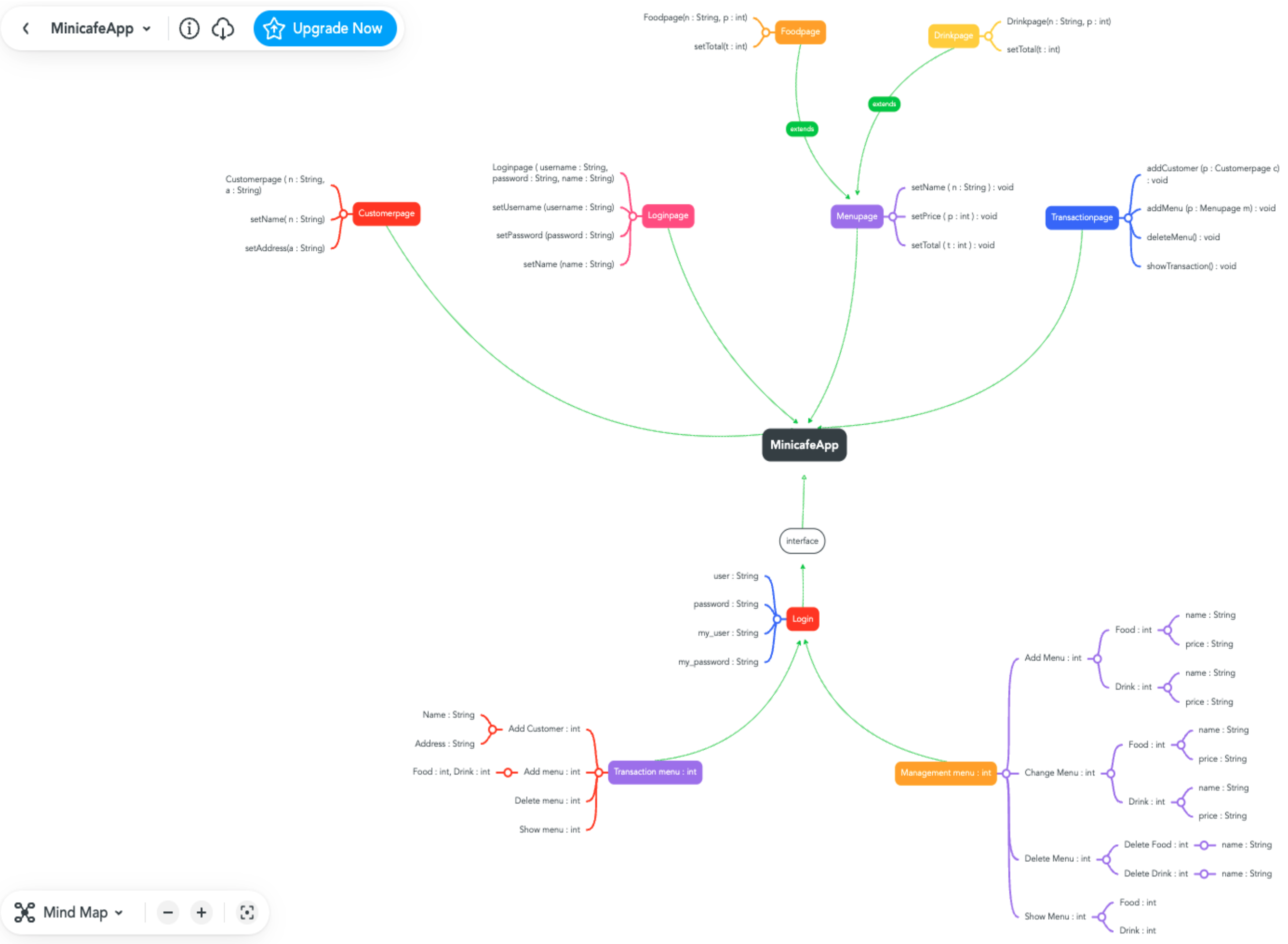
I. Project Description

This is a simple project created with the function to implement the workings in a mini cafe or what we are familiar with as Warkop in the form of a simple application.

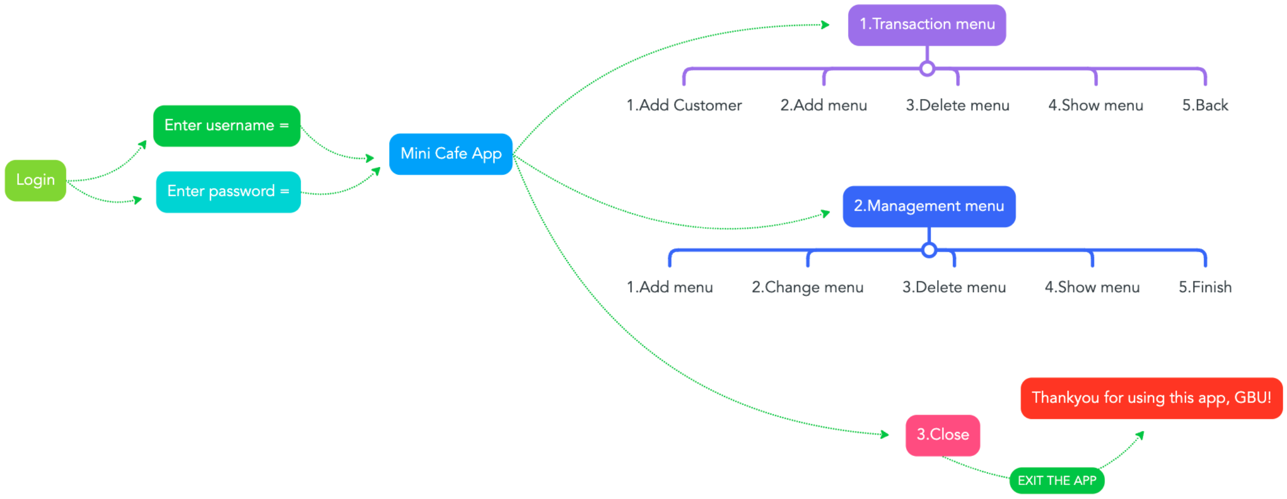
As someone who likes to hang out, I was inspired by those who have a coffee shop business, and I hope that this project can help and make roadside coffee shops a little more sophisticated.

II. Solution Design

- MinicafeApp Flowchart

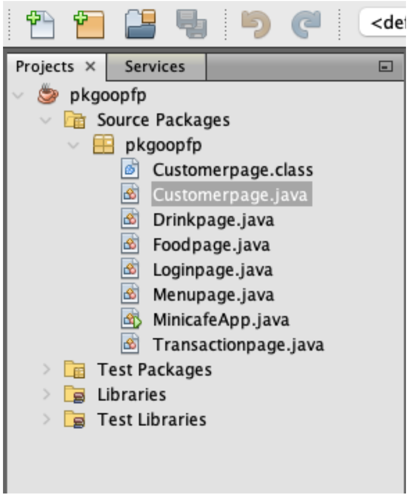


- Application Flow

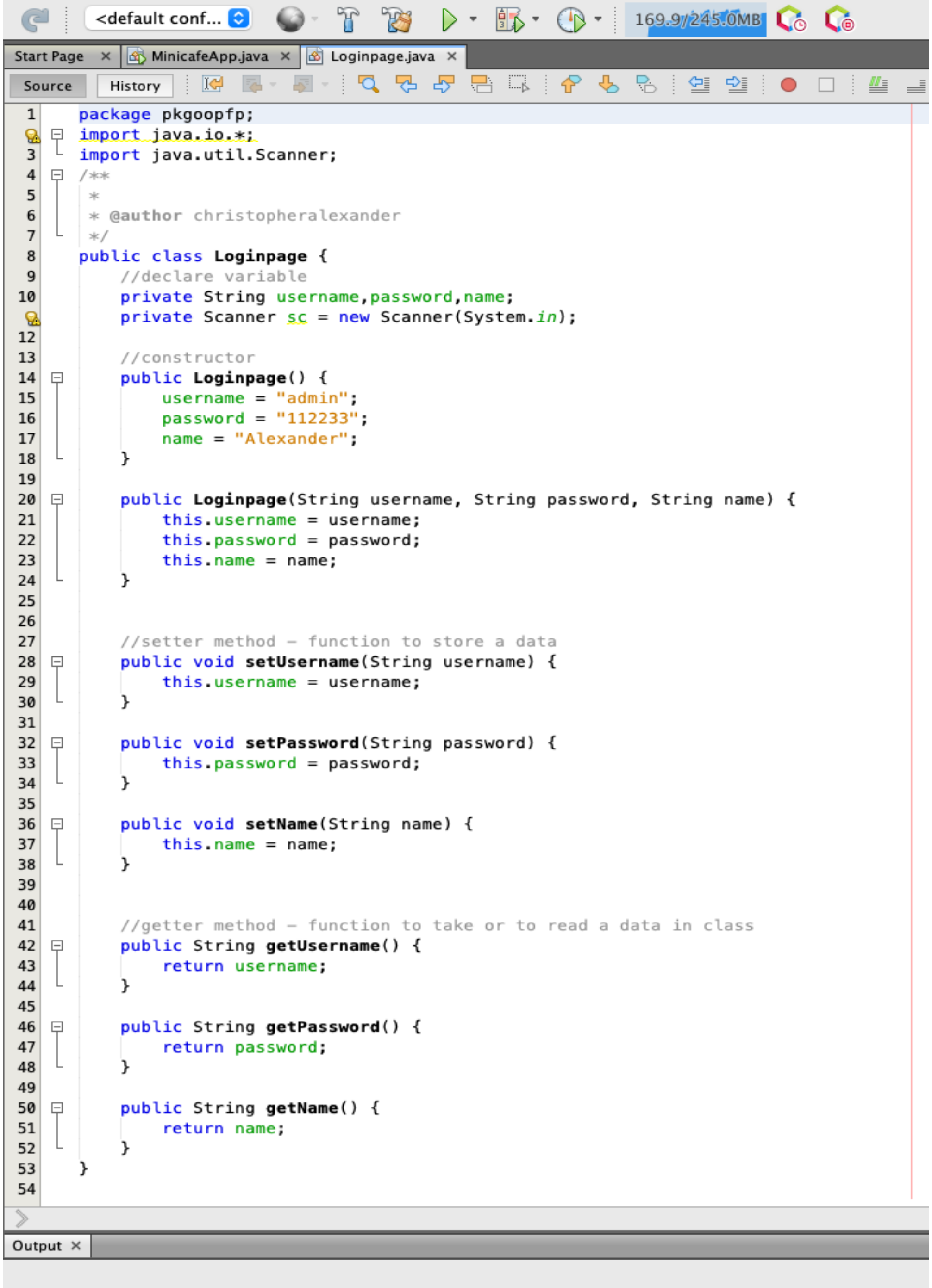


- ☐ **Login - After running the application, you will be bring to the login page which usually an application does.**
- ☐ **Next, you will have to fill the username and password, so you can login to the application.**
- ☐ **After u login, you can now use any features or options in the app, First you can choose 3 main options which is Transaction and Management menu, and also Close for closing the application.**

III. Code Explanation



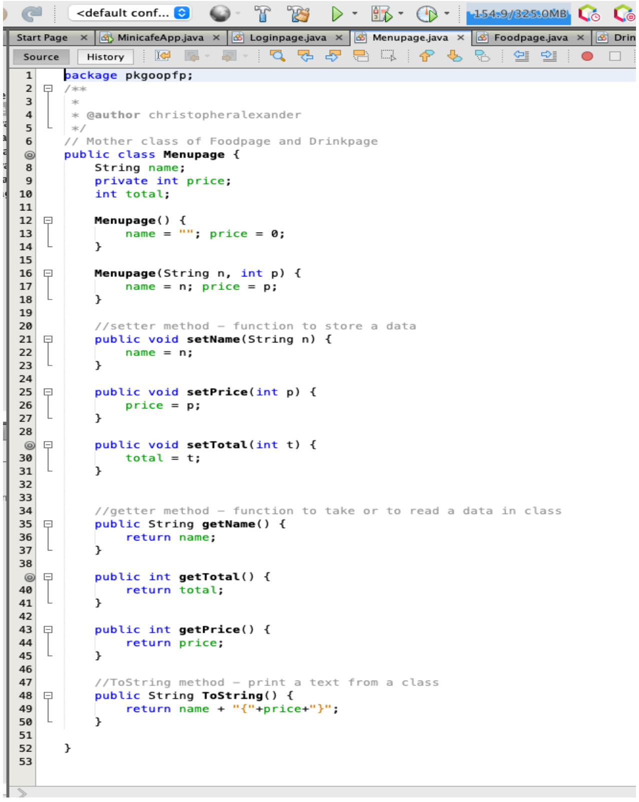
Project Structure - In the picture above, is the structure of the code. All the java classes were in the same package, which is pkgoopfp.



```java
package pkgoopfp;
import java.io.*;
import java.util.Scanner;
/**
 *
 * @author christopheralexander
 */
public class Loginpage {
    //declare variable
    private String username,password,name;
    private Scanner sc = new Scanner(System.in);

    //constructor
    public Loginpage() {
        username = "admin";
        password = "112233";
        name = "Alexander";
    }

    public Loginpage(String username, String password, String name) {
        this.username = username;
        this.password = password;
        this.name = name;
    }


    //setter method - function to store a data
    public void setUsername(String username) {
        this.username = username;
    }

    public void setPassword(String password) {
        this.password = password;
    }

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


    //getter method - function to take or to read a data in class
    public String getUsername() {
        return username;
    }

    public String getPassword() {
        return password;
    }

    public String getName() {
        return name;
    }
}

```

Loginpage.java

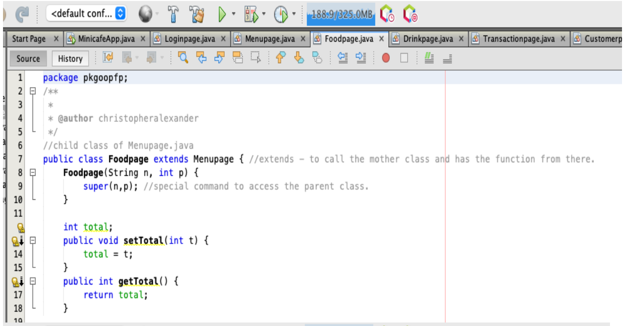
- The following is a Loginpage class which contains several attributes/variables, default constructor, and a setter and getter method. The functions in this class will be used in the application login feature which requires the user to enter a username and password in order to login to the application.



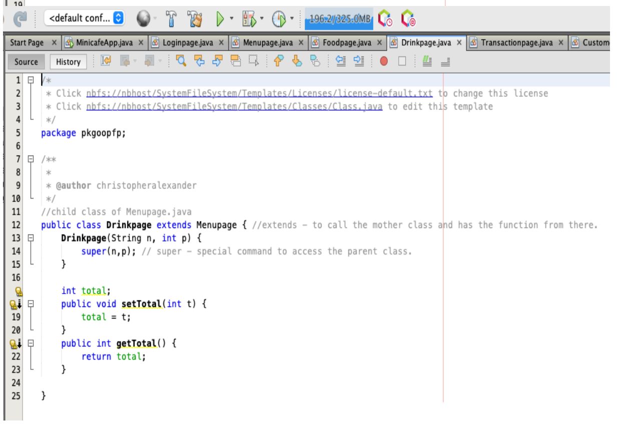
```java
package pkgoopfp;
/**
 *
 * @author christopheralexander
 */
// Mother class of Foodpage and Drinkpage
public class Menupage {
    String name;
    private int price;
    int total;

    Menupage() {
        name = ""; price = 0;
    }

    Menupage(String n, int p) {
        name = n; price = p;
    }

    //setter method - function to store a data
    public void setName(String n) {
        name = n;
    }

    public void setPrice(int p) {
        price = p;
    }

    public void setTotal(int t) {
        total = t;
    }


    //getter method - function to take or to read a data in class
    public String getName() {
        return name;
    }

    public int getTotal() {
        return total;
    }

    public int getPrice() {
        return price;
    }

    //ToString method - print a text from a class
    public String ToString() {
        return name + "{"+price+"}";
    }

}
```

Menupage.java

- The following is a Loginpage class that contains several attributes/variables, default constructor, setter and getter method, and also Tostring method. All functions in this class will be used in the menu features. The function in it will be used to determine the name of the menu and the price, as well as the total. This class is the parent class of Foodpage.java and Drinkpage.java



```java
package pkgoopfp;
/**
 *
 * @author christopheralexander
 */
//child class of Menupage.java
public class Foodpage extends Menupage { //extends - to call the mother class and has the function from there.
    Foodpage(String n, int p) {
        super(n,p); //special command to access the parent class.
    }

    int total;
    public void setTotal(int t) {
        total = t;
    }
    public int getTotal() {
        return total;
    }
```



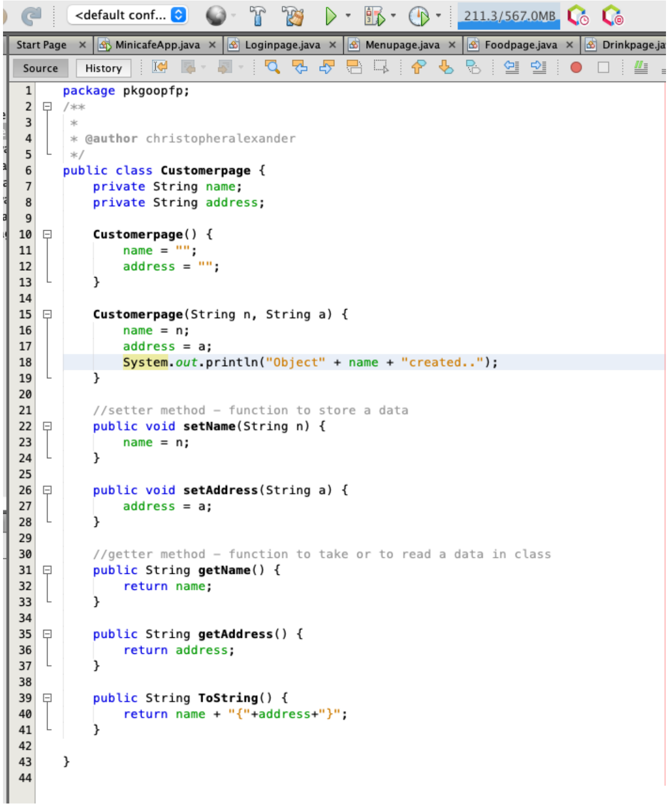
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package pkgoopfp;

/**
 *
 * @author christopheralexander
 */
//child class of Menupage.java
public class Drinkpage extends Menupage { //extends - to call the mother class and has the function from there.
    Drinkpage(String n, int p) {
        super(n,p); // super - special command to access the parent class.
    }

    int total;
    public void setTotal(int t) {
        total = t;
    }
    public int getTotal() {
        return total;
    }

}
```

Foodpage.java & Drinkpage.java

- Both Foodpage and Drinkpage class extends the Menupage class so that they has the function from there. We just need to add some missing variables. Both class were called the child class.

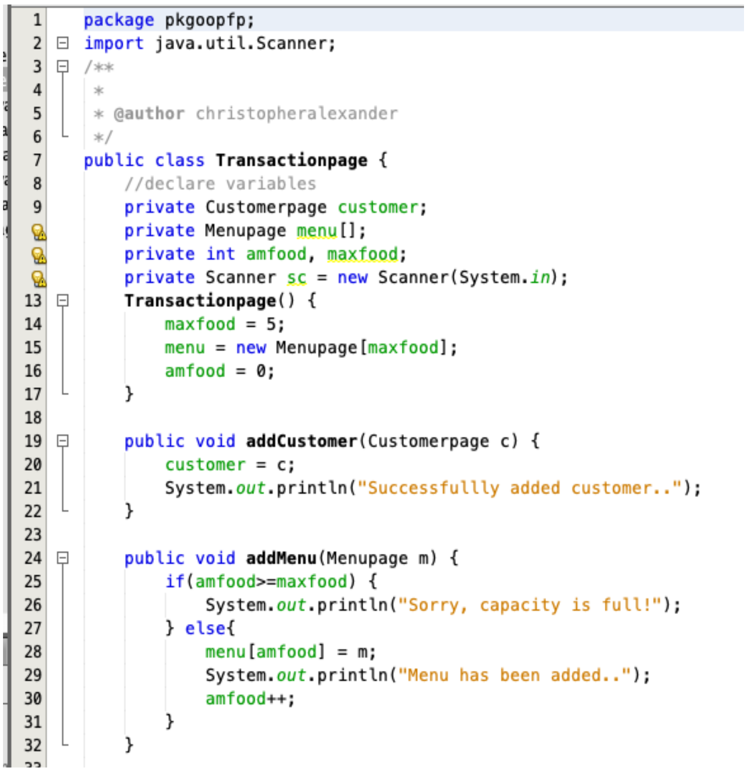
Customerpage.java

- The following is a Customerpage class that contains several attributes/variables, default constructor, setter and getter method, and also Tostring method. This class will be used to set customers and their identities in the Transaction menu.

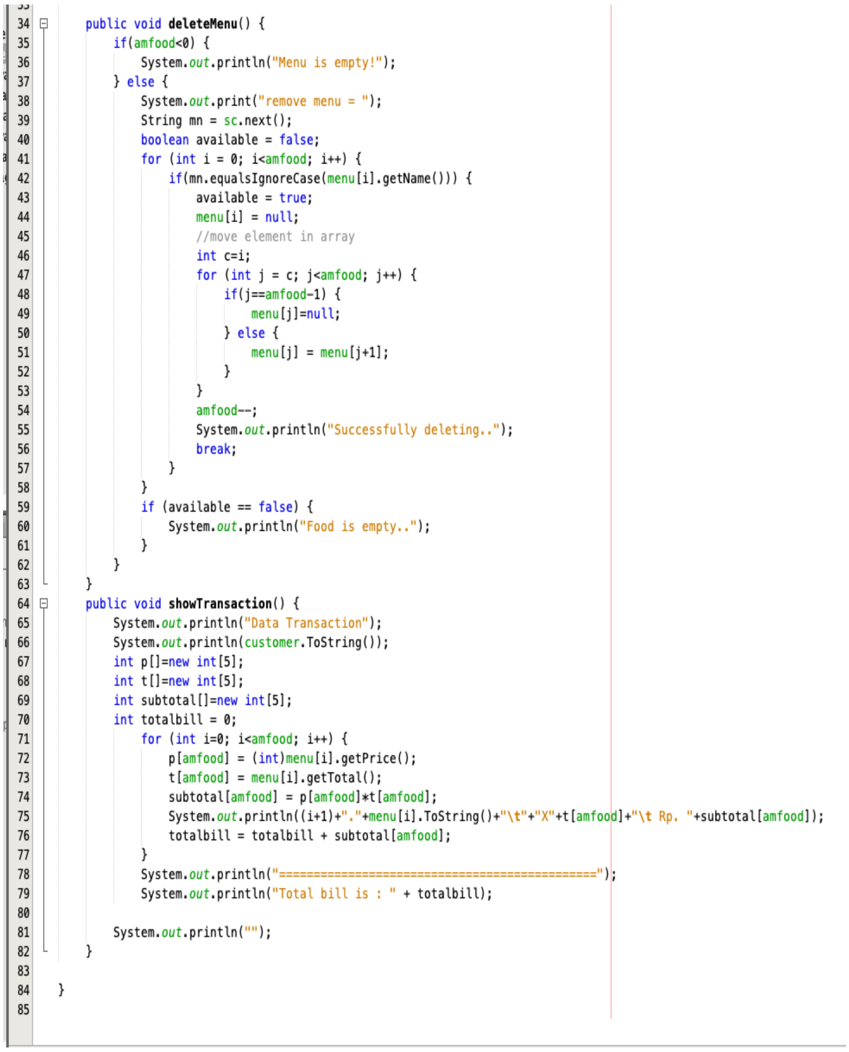


```java
package pkgoopfp;
/**
 *
 * @author christopheralexander
 */
public class Customerpage {
    private String name;
    private String address;

    Customerpage() {
        name = "";
        address = "";
    }

    Customerpage(String n, String a) {
        name = n;
        address = a;
        System.out.println("Object" + name + "created..");
    }

    //setter method - function to store a data
    public void setName(String n) {
        name = n;
    }

    public void setAddress(String a) {
        address = a;
    }

    //getter method - function to take or to read a data in class
    public String getName() {
        return name;
    }

    public String getAddress() {
        return address;
    }

    public String ToString() {
        return name + "{"+address+"}";
    }

}
```

Transactionpage.java



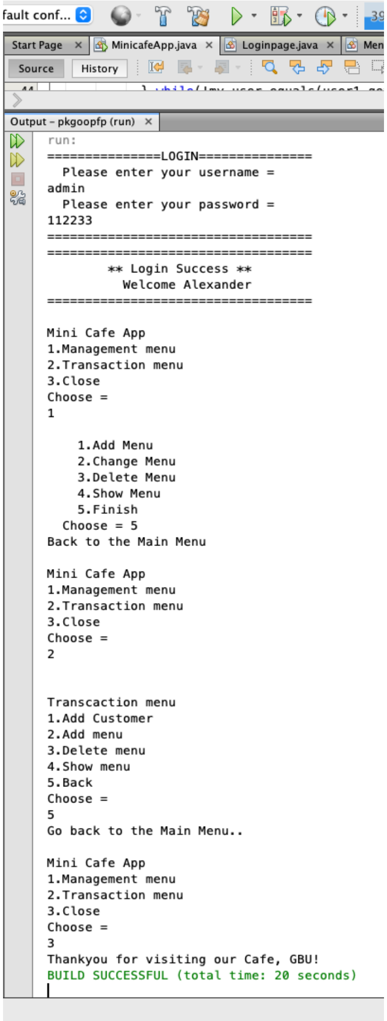
```java
package pkgoopfp;
import java.util.Scanner;
/**
 *
 * @author christopheralexander
 */
public class Transactionpage {
    //declare variables
    private Customerpage customer;
    private Menupage menu[];
    private int amfood, maxfood;
    private Scanner sc = new Scanner(System.in);
    Transactionpage() {
        maxfood = 5;
        menu = new Menupage[maxfood];
        amfood = 0;
    }

    public void addCustomer(Customerpage c) {
        customer = c;
        System.out.println("Successfullly added customer..");
    }

    public void addMenu(Menupage m) {
        if(amfood>=maxfood) {
            System.out.println("Sorry, capacity is full!");
        } else{
            menu[amfood] = m;
            System.out.println("Menu has been added..");
            amfood++;
        }
    }
```



```java
    public void deleteMenu() {
        if(amfood<0) {
            System.out.println("Menu is empty!");
        } else {
            System.out.print("remove menu = ");
            String mn = sc.next();
            boolean available = false;
            for (int i = 0; i<amfood; i++) {
                if(mn.equalsIgnoreCase(menu[i].getName())) {
                    available = true;
                    menu[i] = null;
                    //move element in array
                    int c=i;
                    for (int j = c; j<amfood; j++) {
                        if(j==amfood-1) {
                            menu[j]=null;
                        } else {
                            menu[j] = menu[j+1];
                        }
                    }
                    amfood--;
                    System.out.println("Successfully deleting..");
                    break;
                }
            }
            if (available == false) {
                System.out.println("Food is empty..");
            }
        }
    }
    public void showTransaction() {
        System.out.println("Data Transaction");
        System.out.println(customer.ToString());
        int p[]=new int[5];
        int t[]=new int[5];
        int subtotal[]=new int[5];
        int totalbill = 0;
            for (int i=0; i<amfood; i++) {
                p[amfood] = (int)menu[i].getPrice();
                t[amfood] = menu[i].getTotal();
                subtotal[amfood] = p[amfood]*t[amfood];
                System.out.println((i+1)+"."+menu[i].ToString()+"\t"+"X"+t[amfood]+"\t Rp. "+subtotal[amfood]);
                totalbill = totalbill + subtotal[amfood];
            }
            System.out.println("================================================");
            System.out.println("Total bill is : " + totalbill);

        System.out.println("");
    }

}
```

- This class was created with the aim of managing the transaction system in the application. In Transaction page, there are several attributes/variables and also several methods/functions such as adding customers, adding and removing menus, and displaying customer transactions which totaling everything that customer has ordered into the bill.

MiniCafeApp.java - Driver



```
run:
===============LOGIN===============
  Please enter your username =
admin
  Please enter your password =
112233
===================================
===================================
        ** Login Success **
          Welcome Alexander
===================================

Mini Cafe App
1.Management menu
2.Transaction menu
3.Close
Choose =
1

    1.Add Menu
    2.Change Menu
    3.Delete Menu
    4.Show Menu
    5.Finish
  Choose = 5
Back to the Main Menu

Mini Cafe App
1.Management menu
2.Transaction menu
3.Close
Choose =
2


Transcaction menu
1.Add Customer
2.Add menu
3.Delete menu
4.Show menu
5.Back
Choose =
5
Go back to the Main Menu..

Mini Cafe App
1.Management menu
2.Transaction menu
3.Close
Choose =
3
Thankyou for visiting our Cafe, GBU!
BUILD SUCCESSFUL (total time: 20 seconds)
```

- This class, is a class called a driver. The function of this class is none other than implementing all the functions that exist in several classes that have been described and displaying them in the form of an executable application. Basically, in this class will display application features similar to those described in the class diagram. There will be a login page and after the user has successfully logged in, there is a main function there, namely the management menu which will function for the user to modify whatever is in the cafe, then there is a transaction menu which will function to enter and modify requests from cafe customers. Finally, there is a close function which will close the application.

IV. Resources

Link of the project including source code and some important files :
https://github.com/JugBones/OOP_FP

References :
https://stackoverflow.com/
https://sourcecodehero.com
https://github.com/topics/restaurant-management?l=java