**Cafeteria Management**



Done By –

KOMAL GAUTAM

Submitted to –

Junie Denny Solomon

Summary

ØThe topic of my project is the Café Management System which contains all types of calculations for billing. It is a tool for vendor-slid billing. A Lis of the Cafe Drinks and Cake types is also in the Front View.

ØAfter Entering the quantity we must have to enable the check box for that product

ØThen it will automatically get added to our count.

ØThe Bill will provide in the text area, for the user’s understanding.

ØAll bills contain unique random numbers for reference.

ØFinally, there are four buttons: total, receipt, reset bill, and exit.

**Introduction**

The bill session will appear more complicated in the cafeteria. I planned to create a straightforward application for cafeteria since I believed that I could make it simple. It would be quite difficult to maintain a massive application created by a large team to develop a cafe, therefore I choose to keep it straightforward.

I began creating the café management system with the aid of Java and NetBeans.

The topic of my project is a method for cafeteria billing. It is a tool for vendor-side billing. A list of the cafe's products will be available on the dashboard.

Just the count needs to be added, and the product needs to be enabled.

The bill is provided in the text area, and the bill number is produced at random.

Also, the bill includes the product's name and quantity.

Finally, there are four buttons: total, receipt, reset bill, and exit.

**Tools used**

Apache NetBeans

**Core language:** Java is a **programming language** and a **platform**. Java is a high level, robust, object-oriented and secure programming language.

Java was developed by Sun Microsystems (which is now the subsidiary of Oracle) in the year 1995. James Gosling is known as the father of Java. Before Java, its name was Oak. Since Oak was already a registered company, so James Gosling and his team changed the name from Oak to Java.

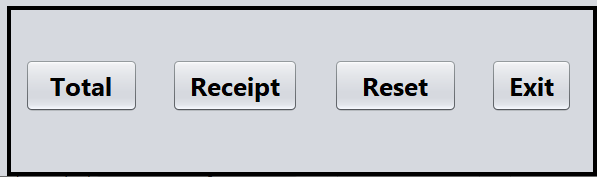
**The main Function used In Java :-**

**Java swing**

The j ava x. swing package provides classes for java swing API such as J Button, J Text Field, J Text Area, J Radio Button, J Checkbox, J Menu, J Colour Chooser etc.

**Java j button**

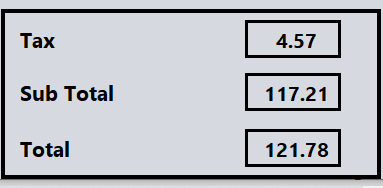
The J Button class is used to create a label button that has platform independent implementation. The application result in some action when the button is pushed. It inherits Abstract Button.

**Java j label**

The object of J Label class is a component for placing text in a container. It is used to display a single line of read only text. The text can be changed by an application but a user cannot edit it directly. It inherits J Component class.

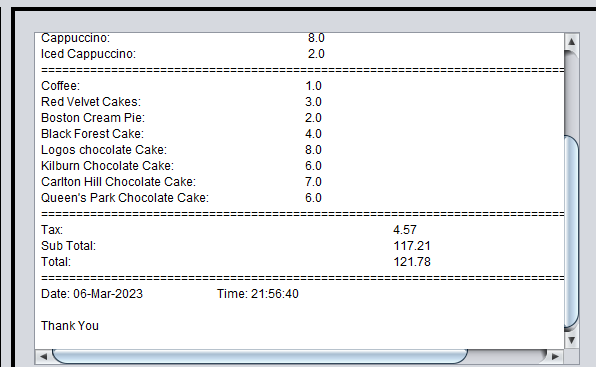
**Java j text field**

The object of a J Text Field class is a text component that allows the editing of a single line text. It inherits J Text Component class.



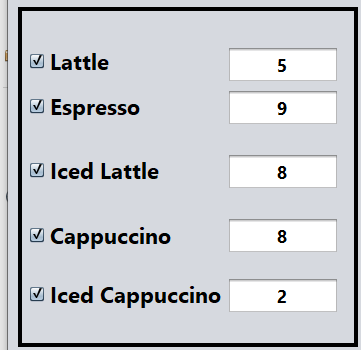
**Java j text Area**

The object of a J Text Area class is a multi line region that displays text. It allows the editing of multiple line text. It inherits J Text Component class.



**Java j Check Box**

The J Check Box class is used to create a checkbox. It is used to turn an option on (true) or off (false). Clicking on a Check Box changes its state from "on" to "off" or from "off" to "on ".It inherits [J Toggle Button](https://www.javatpoint.com/java-jtogglebutton) class.



**Java J Panel**

The J Panel is a simplest container class. It provides space in which an application can attach any other component. It inherits the J Components class.

It doesn't have title bar.

**Java J Frame**

The java x .swing .J Frame class is a type of container which inherits the java. Awt . Frame class. J Frame works like the main window where components like labels, buttons, text fields are added to create a GUI.

Unlike Frame, J Frame has the option to hide or close the window with the help of set Default Close Operation(int) method.

**Java Action Listener interface**

The Java Action Listener is notified whenever you click on the button or menu item. It is notified against Action Event. The Action Listener interface is found in java . awt .event [package](https://www.javatpoint.com/package). It has only one method: action Performed().

**Java mouse Listener interface**

The Java Mouse Listener is notified whenever you change the state of mouse. It is notified against Mouse Event. The Mouse Listener interface is found in java. Awt . event package. It has five methods.

**Java Layout manager**

The Layout Managers are used to arrange components in a particular manner. The **Java Layout Managers** facilitates us to control the positioning and size of the components in GUI forms. Layout Manager is an interface that is implemented by all the classes of layout managers. There are the following classes that represent the layout managers:

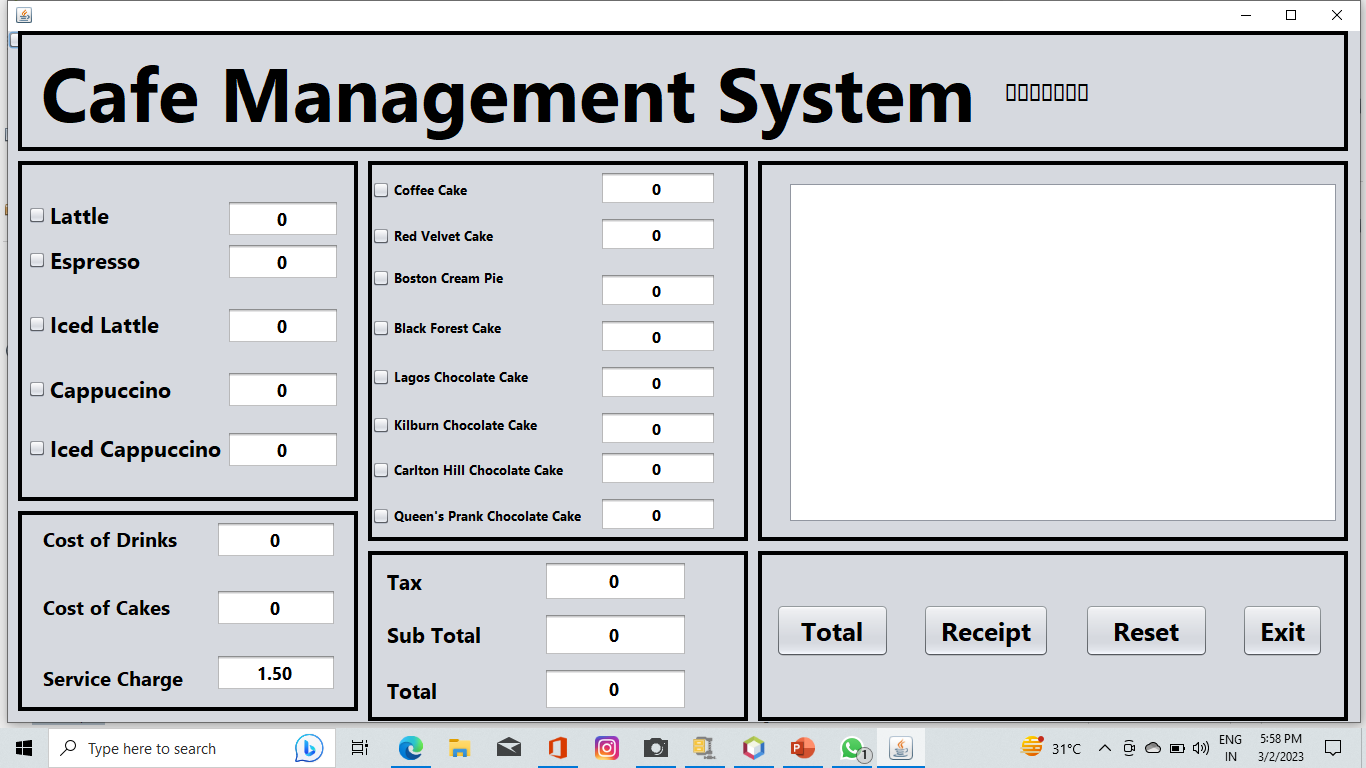
1. Java . awt .Border Layout
2. Java . awt .Flow Layout
3. Java . awt . Grid Layout
4. Java . awt. Card Layout
5. java . awt . Grid Bag Layout
6. java x . swing . Box Layout
7. java x . swing .Group Layout
8. java x . swing . Scroll Pane Layout

9.javax . swing. Spring Layout etc.

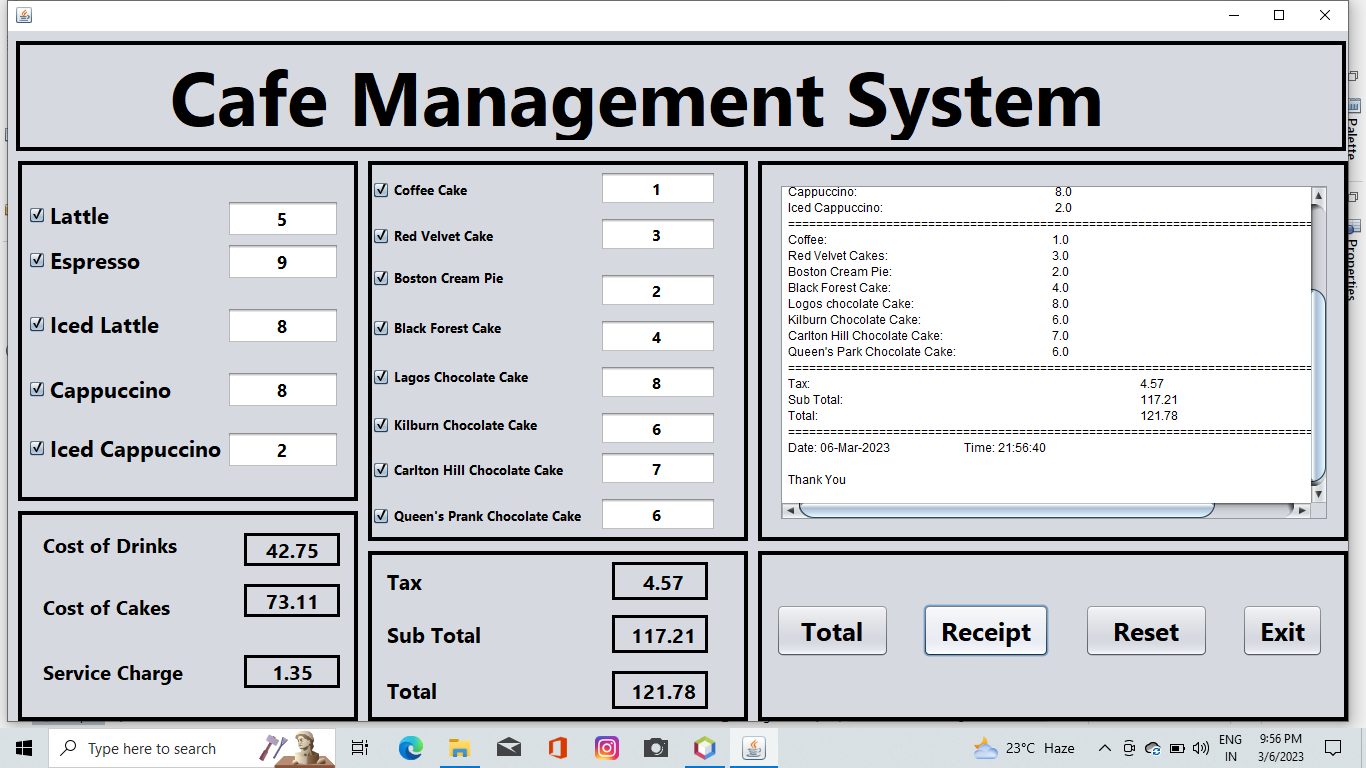
**java Date and time**

The time, util, and .text packages contains classes for representing date and time. Following classes are important for dealing with date in Java.

**Application preview**



**Application preview with output**



**Future work**

In the future, I'll extend the program to connect to a database, store each segment, and create a report page for this project to means of responding.