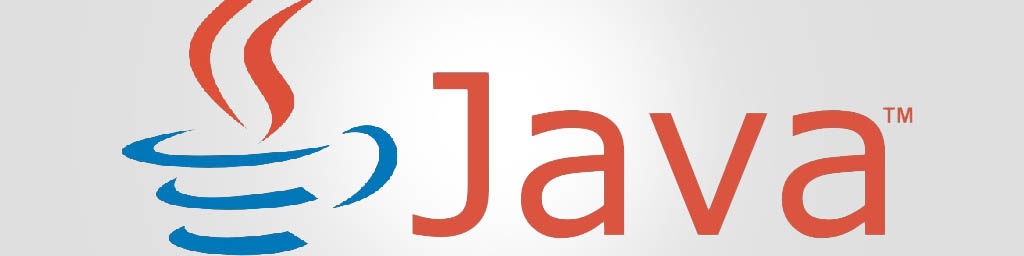
**Point of Sale (POS)**





Made by **Ajinkya Satkar**

Project co-ordinator:-  
 **Junie Denny Solomon**

## Ø **Summary**

The Point Of Sale System, is made entirely out of Java using Jframe with all of the functions that stores need to improve customer satisfaction.

This article on how to make a pos system in java shows us how to improve our skills and logic, which is important for using the Java programming language, which is the most popular and widely used programming language in many businesses.

**Introduction**

. The main job of this Point of Sale System is to handle sales and payments at the point of sale.

The **POS in Java** was made using Java programming and the NetBeans IDE (Integrated Development Environment). This project was created using a graphical user interface (GUI).

This project lets us take payments from customers and keep track of all the sales from the past months, days, or even weeks. The NetBeans IDE (Integrated Development Environment) is used for front-end design.

**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.

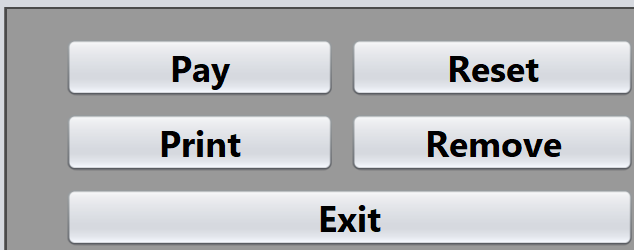
**The main Function used In java :-**

**Java swing**

The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

**Java jbutton**

The JButton class is used to create a labeled button that has platform independent implementation. The application result in some action when the button is pushed. It inherits AbstractButton class.

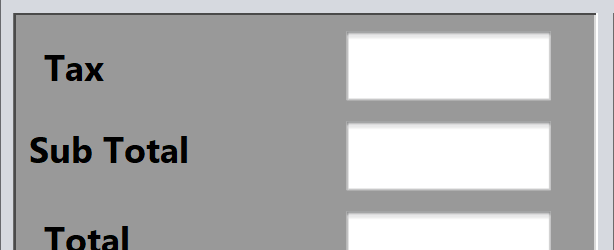


**Java label**

The object of JLabel 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 JComponent class.

**Java jtextfield**

The object of a JTextField class is a text component that allows the editing of a single line text. It inherits JTextComponent class.



**Java jtextArea**

The object of a JTextArea class is a multi line region that displays text. It allows the editing of multiple line text. It inherits JTextComponent class.



**Java jCheckBox**

The JCheckBox class is used to create a checkbox. It is used to turn an option in it Cash, Master card, and Visa-card for payment. Clicking on a CheckBox changes its state from " Cash, Master card, and Visa-card for payment " to "Payment ".It inherits Payment class.



**Java JPanel**

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

It doesn't have title bar.

**Java JFrame**

The javax.swing.JFrame class is a type of container which inherits the java.awt.Frame class. JFrame works like the main window where components like labels, buttons, textfields are added to create a GUI.

Unlike Frame, JFrame has the option to hide or close the window with the help of setDefaultCloseOperation(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**

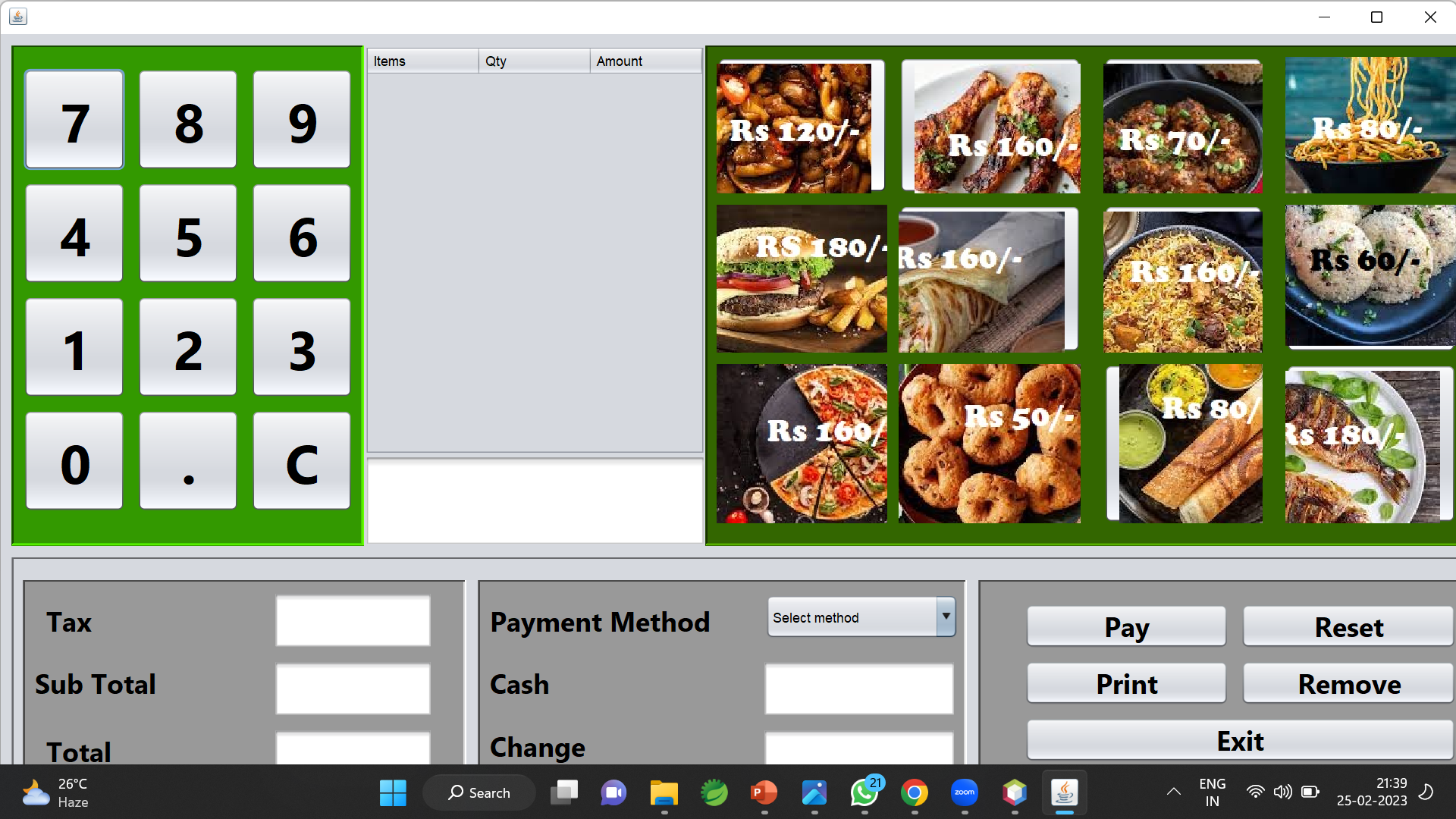
The Java MouseListener is notified whenever you change the state of mouse. It is notified against MouseEvent. The MouseListener interface is found in java.awt.event package. It has five methods.

**Java Layout manager**

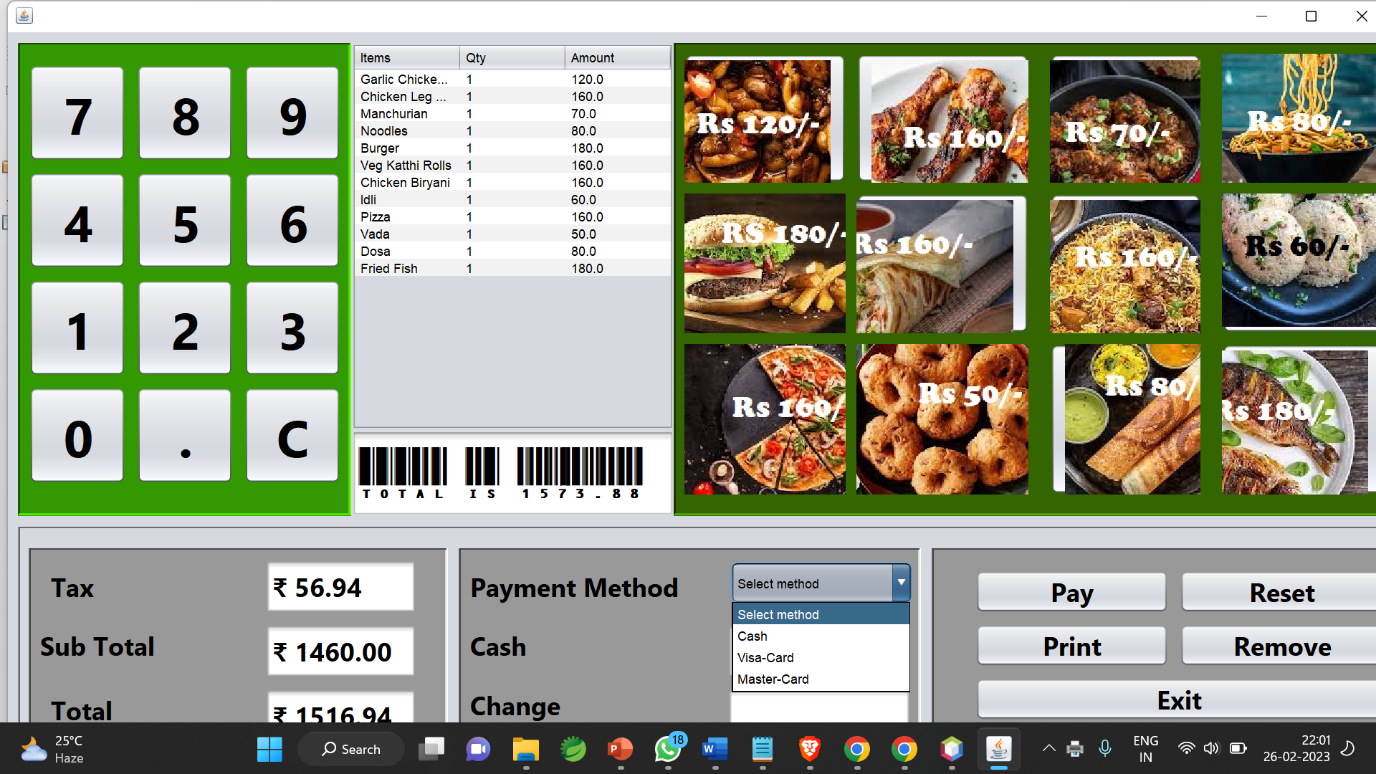
The LayoutManagers are used to arrange components in a particular manner. The **Java LayoutManagers** facilitates us to control the positioning and size of the components in GUI forms. LayoutManager is an interface that is implemented by all the classes of layout managers. There are the following classes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

1. java.awt.BorderLayout
2. java.awt.FlowLayout
3. java.awt.GridLayout
4. java.awt.CardLayout
5. java.awt.GridBagLayout
6. javax.swing.BoxLayout
7. javax.swing.GroupLayout
8. javax.swing.ScrollPaneLayout
9. javax.swing.SpringLayout etc.

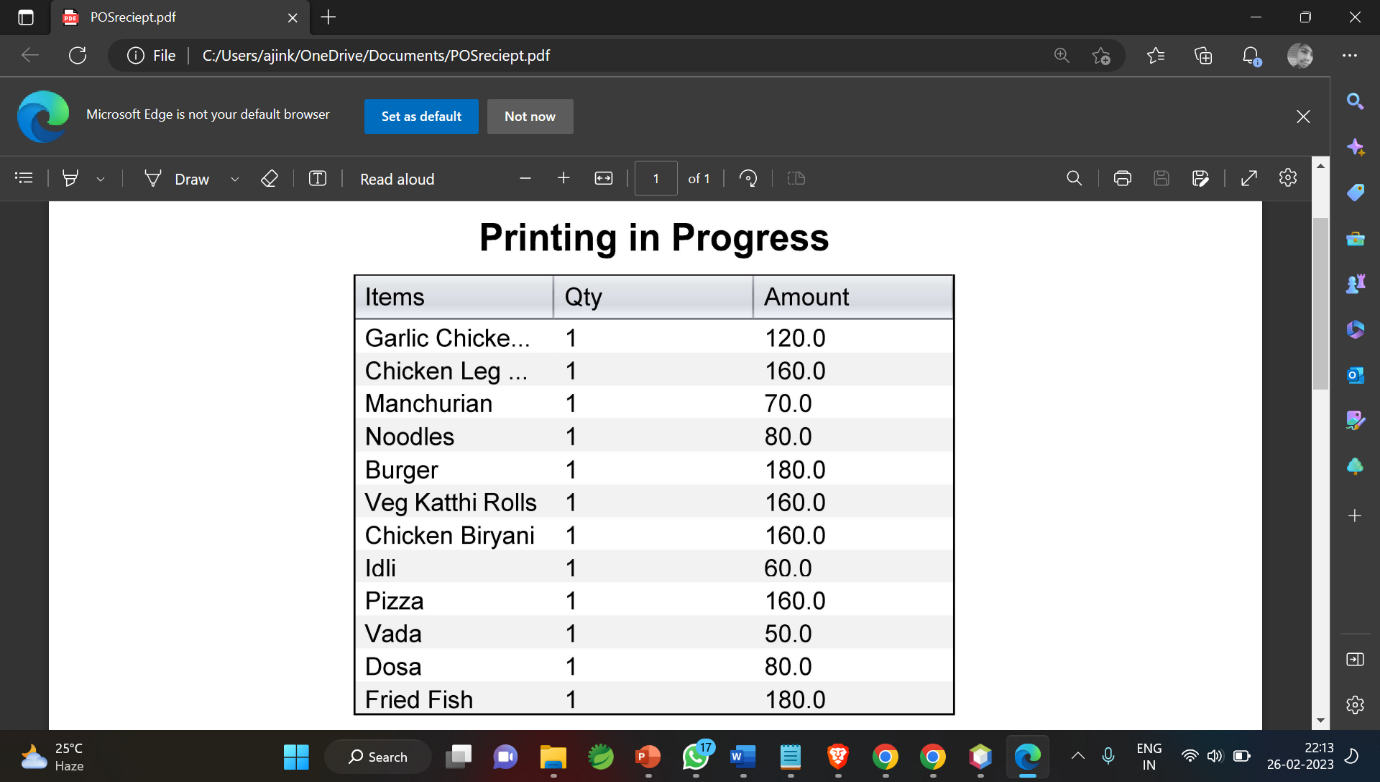
**Application preview**



**Application preview with output**



**Receipt**

****

**Future work**

POS systems will be shaped by customer expectations for personalized, seamless experiences across channels.