**JAVA PROJECT REPORT**

(Project Term January-May 2023)

## *(Note It Down - Notepad Clone)*

Submitted by

**Shanny kumar singh**

**Registration Number :12111009**

**Project Group Number ………….**

**Course Code- CSE310**

Under the Guidance of

**(Dr.A.Ranjith Kumar)**

# School of Computer Science and Engineering



**DECLARATION**

We hereby declare that the project work entitled (“*Note It Down - Notepad Clone*”) is an authentic record of our own work carried out as requirements of Capstone Project for the award of B.Tech degree in CSE from Lovely Professional University, Phagwara, under the guidance of (Dr.A.Ranjith Kumar), during January to April 2023. All the information furnished in this capstone project report is based on our own intensive work and is genuine.

Project Group Number: …………

Name of Student 2: Shanny Kumar Singh

Registration Number: 12213864

(Shanny Kumar Singh)

Date:20-04-2023

**TABLE OF CONTENTS**

Inner first page………………………………………………………………………..(1)

Declaration…...…………………………………………………………………….....(2)

Table of Contents……………………...……………………………………………...(3)

Abstract ……………………………………………………………………………….(4)

1. **Introduction ……..…………………………………………………………….5**
2. **Survey of Existing System ………………………………………………….…6**
3. **Technology …….……………………………………………………………...7**
4. **Methodology ……………………………………………………………………8**
5. **Flowchart for Notepad .……………………………………………………..…9**
6. **Results …………………………………………………………………………10**
7. **Conclusion……… ……………………………………………………………..14**
8. **Reference ………..……………………………………………………………..14**
9. **Appendix ………..……………………………………………………………...15**

**ABSTRACT**

The application is a text editor in Java. This text editor developed in Java platform is a replication of the word editors. We all are familiar with and which we use quite often on a daily basis. The only difference being dad. This editor has been created using Java for the Front air interface. The text edited in the editor is stored in the desired location.

This report is for the detailed study done by us to learn Java by developing Notepad. This application contains some basic features like open file, save file, edit file, print, find and replace text.

#### **INTRODUCTION**

This application is a text-editor in JAVA. This text editor developed in a Java platform, is a replication of the word editors we all are familiar with and, which we use quite often on a daily basis. The text edited in the editor is stored in the desired location. Notepad is a very basic text editor that has been part of windows for a very long time. It is very excellent for writing work, relatively short text documents that you want to save in plain text and that is not all you can do with it.

This application provides us with the ability to create, edit and save files containing text. It also provides us to print this text file. Java provide an user friendly environment which enables selection of various controls like labels, buttons, textbox, etc. and dragging and dropping them at desired position to give a user friendly environment. In designing the software, ease of understanding is maintained to ensure that a person with a little knowledge of computer can work efficiently.

**1.1 Objective-**

The main objective was to study Java programming by developing a notepad. A notepad i.e. a text-editor is a computer program that lets a user enter, change, store, and usually print text (characters and numbers).

### **2. Survey of Existing System**

A useful text-editor is an essential component of any personal computer. Different companies and people use different types of text editors. And there are slight variations in this text editors are relative to one another. But they all perform the same function of editing and viewing a text file.

Every day we need to make notes, compose document, and record vital pieces of information. In existing system multiple tabbed opening feature is not available. We cannot change background colour and foreground colour of text editor as per users choice. Existing system is not user friendly because user cannot customize software. But there are some features in notepad like open file, save file, edit, font, cut, copy, paste, find, etc. It can be used by anyone for viewing and editing a text file. It is a user friendly application.

**Features of Notepad are:**

* Easy to use: A notepad is simple, handy, fast and accurate tool for editing a text file. This means that users can use this app in any situation where quick viewing and editing is needed.
* Offers speedy operations: Accuracy and speed are very crucial.
* It is reliable: The reliability of Notepad cannot be questioned. Many Web developers and office workers have been using it for quite some time now.
* It can be used for any type of text files: With a notepad, you can view an edit any type of text file on your laptop for desktop computers.
* The use of Java helps to keep the information and restore it easily. Apart from this, it has security manager that defines the access of classes. Java is straight- forward to use the right view edit and programming languages.

## 3. TECHNOLOGY

## JFrame:

JFrame is a class in the Java Swing library that provides a basic window for creatinggraphicaluser interfaces (GUI) in Java. It is a top-level container that can be used to hold other Swing components such as buttons, labels, text fields, and other GUI elements.

JFrame provides a platform-independent way to create GUI applications that can run on

differentoperating systems. It has a wide range of functionality such as displaying images, menus, and toolbars. JFrame can also be customized with different colors, fonts, and sizes to

create visually appealing user interfaces.

JFrame inherits from the java.awt. Frame class, which is part of the Abstract Window

Toolkit (AWT) in Java. However, JFrame is preferred over Frame because it is built.

on top of Swing, which provides a richer set of components and is more flexible and

customizable than AWT.

To create a JFrame in Java, developers must first import the necessary classes from the Swing

library. Then, they can create a new JFrame object and add other components to it as needed.

Developers can also set properties of the JFrame such as its size, title, and default close operation.

Overall, JFrame is a versatile class in the Java Swing library that provides a basic window for creating GUI applications in Java. Its flexibility and customizability make it an essential tool for developers who want to create visually appealing and use**r** friendly application.

**4. METHODOLOGY**

# 4.1 Requirements

The first step is gathering our requirements of the project. The functions and the modules and the necessary for the development of the project.

# 4.2 Functionality

In this step we will discuss the different function used in the program and how they are working

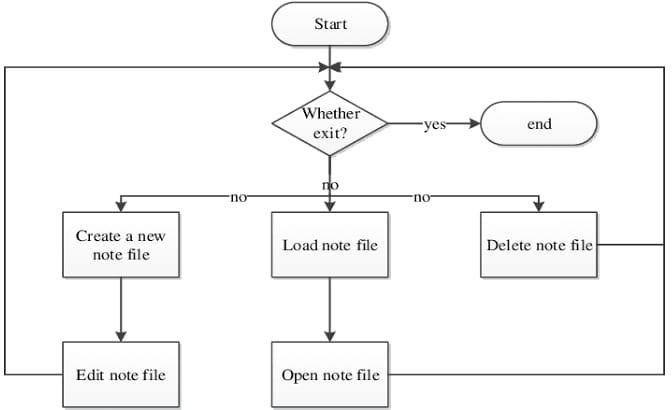
* **File:** In this section drop down menu is available in which we can perfom various operation listed bellow

Table

Description automatically generated

* **Edit:** In this section drop down menu is available in which we can perfom various operation listed bellow

**5. Flowchart for Notepad**



**6. Results:**

**6.1 File section**

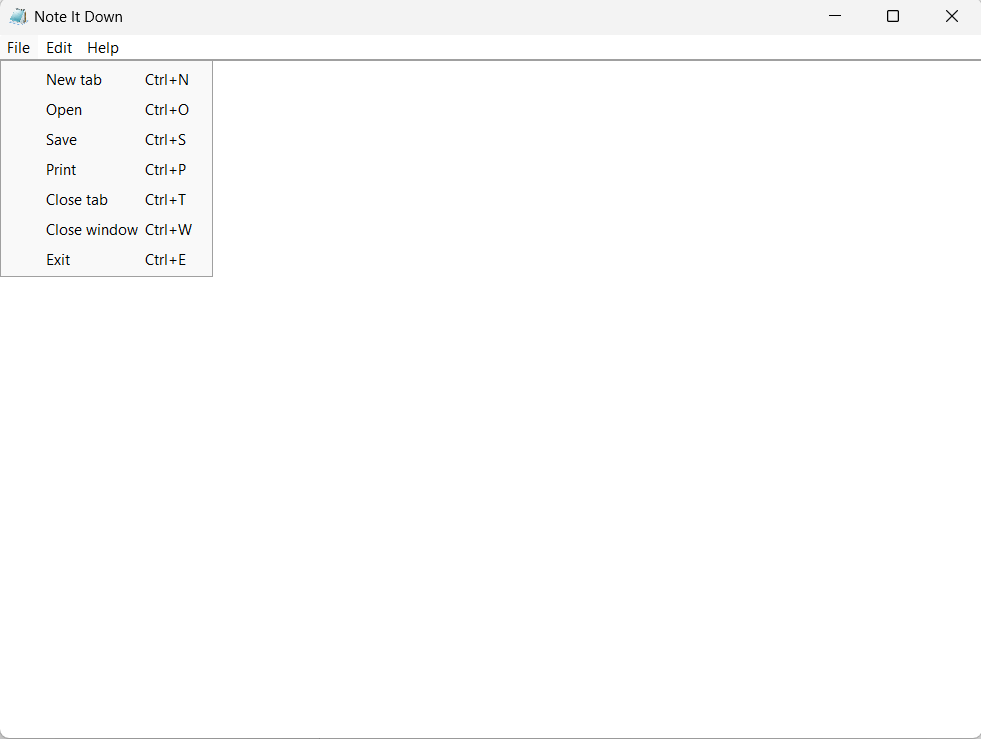


Figure -1

**6.2 Edit Section**

Graphical user interface, application, Word

Description automatically generated

Figure -2

**6.3 About section**

Graphical user interface, application

Description automatically generated

Figure -3

**6.4 Writing words in Note in Down**

Graphical user interface, text, application

Description automatically generated

Figure -4

**6.5 Selecting the word**

Graphical user interface, text, application, Word

Description automatically generated

Figure -5

**6.6 Copying the word and pasting**

Graphical user interface, text, application

Description automatically generated

Figure -6

**6.7 Saving the file in C Drive**

Graphical user interface, text, application

Description automatically generated

Figure -7

**6.8 Printing the page:**

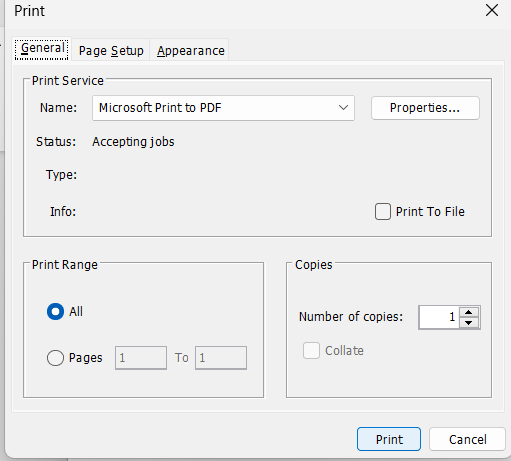


Figure -8

**7. Conclusion**

Here we conclude our lines of our mini project on the topic ‘*Note It Down - Notepad Clone*’ with extreme satisfaction and contentment. This report consists brief definition of Notepad together with its features and functions.

Added to this, this report contains the basic description to create, edit, save, delete and exit from file through JAVA program. It is user friendly, which can be used by the user to perform desired operations. It also includes practical implementation of text editors through complex JAVA program.

**8. Reference**

## [www.google.co.in](http://www.google.co.in)

## <https://www.javatpoint.com>

## https://chat.openai.com/chat/b8a39b1b-6f36-4678-9e98-b6fb7310d346

**9. Appendix**

**9.1 About.java**

import javax.swing.\*;

import java.awt.\*;

public class About extends JFrame{

    About(){

        setTitle("About Note It Down Application");

        setBounds(100, 100, 800, 600);

        setResizable(false);

        setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

        setLayout(null);

        ImageIcon icon = new ImageIcon(getClass().getResource("notepad.png"));

        setIconImage(icon.getImage());

        JLabel iconLabel= new JLabel(new ImageIcon(getClass().getResource("windows14.png")));

        iconLabel.setBounds(40, 80, 800, 100);

        add(iconLabel);

        JLabel textLabel = new JLabel("<html>Note It Down<br><br> All rights reserved<br><br>Note It Down is a word processing program, <br>which allows changing of text in a computer file.<br>Note It Down is a simple text editor for basic text-editing program which enables computer users to create documents. </html>");

        textLabel.setFont(new Font("SAN\_SERIF", Font.PLAIN, 18));

        textLabel.setBounds(150, 130, 500, 300);

        add(textLabel);

    }

    public static void main(String[] args){

        new About().setVisible(true);

    }

}

**9.2 Notepad.java**

import java.awt.\*;

import java.awt.event.\*;

import java.awt.print.PrinterException;

import javax.swing.\*;

import java.io.\*;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.filechooser.\*;

public class Notepad extends JFrame implements ActionListener {

    JMenuBar menubar = new JMenuBar();

    JMenu file = new JMenu("File");

    JMenu edit = new JMenu("Edit");

    JMenu help = new JMenu("Help");

    JMenuItem newFile = new JMenuItem("New tab");

    JMenuItem openFile = new JMenuItem("Open");

    JMenuItem saveFile = new JMenuItem("Save");

    JMenuItem print = new JMenuItem("Print");

    JMenuItem exit = new JMenuItem("Exit");

    JMenuItem cut = new JMenuItem("Cut");

    JMenuItem copy = new JMenuItem("Copy");

    JMenuItem paste = new JMenuItem("Paste");

    JMenuItem selectall = new JMenuItem("Select All");

    JMenuItem closeTab = new JMenuItem("Close tab");

    JMenuItem closeWindow = new JMenuItem("Close window");

    JMenuItem about = new JMenuItem("About");

    JTextArea textArea = new JTextArea();

    Notepad() {

        setTitle("Note It Down");

        setBounds(100, 100, 800, 600);

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        ImageIcon icon = new ImageIcon(getClass().getResource("notepad.png"));

        setIconImage(icon.getImage());

        setJMenuBar(menubar);

        menubar.add(file);

        menubar.add(edit);

        menubar.add(help);

        file.add(newFile);

        file.add(openFile);

        file.add(saveFile);

        file.add(print);

        file.add(closeTab);

        file.add(closeWindow);

        file.add(exit);

        edit.add(cut);

        edit.add(copy);

        edit.add(paste);

        edit.add(selectall);

        help.add(about);

        JScrollPane scrollpane = new JScrollPane(textArea);

        add(scrollpane);

        textArea.setFont((new Font(Font.SANS\_SERIF, Font.PLAIN, 20)));

        scrollpane.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL\_SCROLLBAR\_NEVER);

        scrollpane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL\_SCROLLBAR\_AS\_NEEDED);

        scrollpane.setBorder(BorderFactory.createEmptyBorder());

        textArea.setLineWrap(true);

        textArea.setWrapStyleWord(true);

        newFile.addActionListener(this);

        openFile.addActionListener(this);

        saveFile.addActionListener(this);

        print.addActionListener(this);

        exit.addActionListener(this);

        cut.addActionListener(this);

        copy.addActionListener(this);

        paste.addActionListener(this);

        selectall.addActionListener(this);

        closeTab.addActionListener(this);

        closeWindow.addActionListener(this);

        about.addActionListener(this);

        newFile.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_N, KeyEvent.CTRL\_DOWN\_MASK));

        openFile.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_O, KeyEvent.CTRL\_DOWN\_MASK));

        saveFile.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_S, KeyEvent.CTRL\_DOWN\_MASK));

        print.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_P, KeyEvent.CTRL\_DOWN\_MASK));

        exit.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_E, KeyEvent.CTRL\_DOWN\_MASK));

        cut.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_X, KeyEvent.CTRL\_DOWN\_MASK));

        copy.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_C, KeyEvent.CTRL\_DOWN\_MASK));

        paste.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_P, KeyEvent.CTRL\_DOWN\_MASK));

        selectall.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_A, KeyEvent.CTRL\_DOWN\_MASK));

        closeTab.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_T, KeyEvent.CTRL\_DOWN\_MASK));

        closeWindow.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_W, KeyEvent.CTRL\_DOWN\_MASK));

        about.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_A, KeyEvent.CTRL\_DOWN\_MASK));

    }

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

        UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());

        new Notepad().setVisible(true);

    }

    @Override

    public void actionPerformed(ActionEvent e) {

        if (e.getActionCommand().equals("New")) {

            textArea.setText(null);

        } else if (e.getActionCommand().equalsIgnoreCase("Save")) {

            JFileChooser fileChooser = new JFileChooser();

            FileNameExtensionFilter textFilter = new FileNameExtensionFilter("Only Text Files(.txt)", "txt");

            fileChooser.setAcceptAllFileFilterUsed(false);

            fileChooser.addChoosableFileFilter(textFilter);

            int action = fileChooser.showSaveDialog(null);

            if (action != JFileChooser.APPROVE\_OPTION) {

                return;

            } else {

                String fileName = fileChooser.getSelectedFile().getAbsolutePath().toString();

                if (!fileName.contains(".txt"))

                    fileName = fileName + ".txt";

                try {

                    BufferedWriter writer = new BufferedWriter(new FileWriter(fileName));

                    textArea.write(writer);

                } catch (IOException ex) {

                    ex.printStackTrace();

                }

            }

        } else if (e.getActionCommand().equalsIgnoreCase("Open")) {

            JFileChooser fileChooser = new JFileChooser();

            FileNameExtensionFilter textFilter = new FileNameExtensionFilter("Only Text Files(.txt)", "txt");

            fileChooser.setAcceptAllFileFilterUsed(false);

            fileChooser.addChoosableFileFilter(textFilter);

            int action = fileChooser.showSaveDialog(null);

            if (action != JFileChooser.APPROVE\_OPTION) {

                return;

            } else {

                String fileName = fileChooser.getSelectedFile().getAbsolutePath().toString();

                if (!fileName.contains(".txt"))

                    fileName = fileName + ".txt";

                try {

                    BufferedWriter writer = new BufferedWriter(new FileWriter(fileName));

                    textArea.write(writer);

                } catch (IOException ex) {

                    ex.printStackTrace();

                }

            }

        } else if (e.getActionCommand().equalsIgnoreCase("Print")) {

            try {

                textArea.print();

            } catch (PrinterException ex) {

                Logger.getLogger(Notepad.class.getName()).log(Level.SEVERE, null, ex);

            }

        } else if (e.getActionCommand().equalsIgnoreCase("Exit")) {

            System.exit(0);

        }else if (e.getActionCommand().equalsIgnoreCase("Close tab")) {

            System.exit(0);

        }

        else if (e.getActionCommand().equalsIgnoreCase("Close window")) {

            System.exit(0);

        }

        else if (e.getActionCommand().equalsIgnoreCase("Cut")) {

            textArea.cut();

        } else if (e.getActionCommand().equalsIgnoreCase("Copy")) {

            textArea.copy();

        } else if (e.getActionCommand().equalsIgnoreCase("Paste")) {

            textArea.paste();

        } else if (e.getActionCommand().equalsIgnoreCase("Select All")) {

            textArea.selectAll();

        } else if (e.getActionCommand().equalsIgnoreCase("About")) {

            new About().setVisible(true);

        }

    }

}