

Project Report on

Text-Editor App

**Submittedby:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Roll**  **No.** | **Reg.No** | **Section** | **Group** |
| Mohd Ahad | 50 | 12115116 | K21QT | 2 |
| Mohd Raise | 20 | 12115470 | K21QT | 1 |
| Md Lucky Ahmad | 31 | 12115110 | K21QT | 1 |

**Submitted to: Dr. Bhimasen moharana**

**DEPARTMENT OF**

**COMPUTER SCIENCE AND ENGINEERING**

*INTRODUCTION*

***A text editor is a tool that allows a user to create and revise documents in a computer****. Though this task can be carried out in other modes, the word text editor commonly refers to the tool that does this interactively. To create a text editor in Java, you can start by creating a Frame that contains a Text Area component for the user to input and edit text. You can then add a Menu Bar that contains menu items for file operations like "New", "Open", "Save", and "Save As". For each menu item, you can add an Action Listener that performs the corresponding file operation.*

*For example, when the "New" menu item is clicked, you can create a new Text Area and set the current file to null. When the "Open" menu item is clicked, you can use a File Chooser to allow the user to select a file to open, and then read the contents of the file into the Text Area.*

*What is text editor and why it is important ?*

*In java, text editor is a simple program that allows you to create, open, and change plain text files. This is not the same as word processors. This is used to make documents that provide well-organized and easy to read text. Also, text editors are more important than word processors like open office because they focus on plain text, while text editors are also often used to write documentation and source code for computer programs.*

**More detailed explanation about this projects**

This simple java text editor was developed and written in java and designed in the java swing toolkit, which uses different menu items to find and replace. We used Text area, Menu Items. All the actions dime by the users are tracked using action Listener.

*Tools*

This project “Text Editor” is software which can edit plain text. It is made using java swings and AWT.

In this project all the frames are designed in swing. Today most programmers use swing. Swing is a set of classes that provides more powerful and flexible GUI components than does the AWT. Swing provides the look and feel of the modern java GUI.

Swing did not exist in the early days of java. Rather, it was a response to deficiencies present in java original GUI subsystem: The abstract window toolkit. The AWT defines a basic set of controls, windows, and dialog boxes that support a usable, but limited graphical interface.

*System development life cycle(SDLC)*

***This is also known as classic life cycle model(or) linear sequential model or waterfall method. This has the following activities.***

1. **System/ information engineering and modeling**
2. **Software requirement analysis**
3. **System analysis and design**
4. **Code generation**
5. **Testing**
6. **Maintenance**

*System information engineering and modeling*

As software is always of a large system, work begins by establishing requirements for all system elements and then allocating some subset of these requirements to software. This system view is essential when software must interface with other elements such as hardware, people and other resource. System is the basic and very critical requirement for the existence of software in any entity. So if the system is not in place, the system should be engineered and put in place. In some cases, to extract the maximum output, the system should be re-engineered and spruced up. Once the ideal system is engineered or tuned, the development team studies the software requirement for the system.

*Software requirement analysis*

*Software Requirement Analysis This is also known as feasibility study. In this phase, the development team visits the customer and studies their system. They investigate the need for possible software automation in the given system. By the end of the feasibility study, the team furnishes a document that holds the different specific recommendations for the candidate system. It also includes the personnel assignments, costs, project schedule, and target dates. The requirements gathering process is intensified and focused specially on software. To understand the nature of the program(s) to be built, the system engineer ("analyst") must understand the information domain for the software, as well as required function, behavior, performance and interfacing. The essential purpose of this phase is to find the need and to define the problem that needs to be solved*

*System analysis and design*

*System Analysis and Design In this phase, the software development process, the software's overall structure and its nuances are defined. In terms of the client/server technology, the number of tiers needed for the package architecture, the database design, the data structure design etc are all defined in this phase. A software development model is created. Analysis and Design are very crucial in the whole development cycle. Any glitch in the design phase could be very expensive to solve in the later stage of the software development. Much care is taken during this phase. The logical system of the product is developed in this phase.*

*Code Generation*

*By the help of java programming we have design a code for text editor.*

*import javax.swing.\*;*

*import javax.swing.event.ChangeEvent;*

*import javax.swing.event.ChangeListener;*

*import javax.swing.filechooser.FileNameExtensionFilter;*

*import javax.swing.plaf.basic.BasicBorders;*

*import java.awt.\*;*

*import java.awt.event.\*;*

*import java.io.File;*

*import java.io.FileNotFoundException;*

*import java.io.PrintWriter;*

*import java.util.Scanner;*

*public class TextEditor extends JFrame implements ActionListener {*

*JTextArea textArea;*

*JScrollPane scrollPane;*

*JSpinner fontSizeSpinner;*

*JLabel fontLabel,backgroundColorChangeLabel;*

*JButton fontColorButton,backgroundColorButton,modeButton,boldTextButton;*

*JComboBox fontbox;*

*JMenuBar menuBar;*

*JMenu fileMenu;*

*JMenuItem openItem,saveItem,exitItem;*

*TextEditor(){*

*this.setDefaultCloseOperation (JFrame.EXIT\_ON\_CLOSE);*

*this.addWindowStateListener (new WindowStateListener () {*

*@Override*

*public void windowStateChanged (WindowEvent e) {*

*// minimized*

*if ((e.getNewState() & Frame.ICONIFIED) == Frame.ICONIFIED){*

*scrollPane.setPreferredSize (new Dimension (480,480));*

*//textArea.setPreferredSize (new Dimension (480,480));*

*// scrollPane.requestFocus ();*

*}*

*// maximized*

*else if ((e.getNewState() & Frame.MAXIMIZED\_BOTH) == Frame.MAXIMIZED\_BOTH){*

*//textArea.setPreferredSize (new Dimension (1450,700));*

*scrollPane.setPreferredSize (new Dimension (1450,700));*

*}*

*}*

*});*

this.getContentPane ().setBackground (new Color (0xEDEDEE));

this.setTitle ("Simpe Text Editor");

this.setSize (600,600);

this.setLayout (new FlowLayout ());

this.setLocationRelativeTo (null);

//text are style and code starts here ----------------------------//

textArea = new JTextArea ();

textArea.setLineWrap (true);

textArea.setWrapStyleWord (true);

textArea.setFont (new Font ("Arial Narrow",Font.PLAIN,20));

textArea.setBackground (new Color (250, 250, 250));

textArea.setText ("write something.......");

textArea.setMargin (new Insets(10,10,10,10));

textArea.getText ();

//Scroll pane code------------------------------------------//

scrollPane = new JScrollPane (textArea);

scrollPane.setPreferredSize (new Dimension (480,480));

scrollPane.setVerticalScrollBarPolicy (ScrollPaneConstants.VERTICAL\_SCROLLBAR\_ALWAYS);

fontLabel = new JLabel ("Font: ");

fontSizeSpinner = new JSpinner ();

fontSizeSpinner.setPreferredSize (new Dimension (50,25));

fontSizeSpinner.setValue (20);

//code to change font size

fontSizeSpinner.addChangeListener (new ChangeListener () {

@Override

public void stateChanged (ChangeEvent e) {

int fontSize = (int)fontSizeSpinner.getValue ();

if(fontSize <5){

fontSize = 5;

fontSizeSpinner.setValue (5);

}

if(fontSize >50){

fontSize = 50;

fontSizeSpinner.setValue (50);

}

textArea.setFont (new Font (textArea.getFont ().getFamily (),textArea.getFont ().getStyle (),fontSize));

}

});

//font colour button

fontColorButton = new JButton ("Color");

fontColorButton.addActionListener (this);

//text bold button----------------------------------------------

boldTextButton = new JButton ("Bold Text");

boldTextButton.addActionListener (this);

//font box

//this will take all fonts in java store it in fonts array

String[] fonts = GraphicsEnvironment.getLocalGraphicsEnvironment ().getAvailableFontFamilyNames ();

fontbox = new JComboBox (fonts);

fontbox.addActionListener (this);

fontbox.setSelectedItem ("Arial");

//-----------------background color change option-----------------//

backgroundColorChangeLabel = new JLabel ("Change Background");

backgroundColorButton = new JButton ("White");

backgroundColorButton.addActionListener (this);

//code to get toggle between night mode

modeButton = new JButton ("Night Mode");

modeButton.addActionListener (this);

//---------MenuBar------------------------------------//

menuBar = new JMenuBar ();

fileMenu = new JMenu ("File");

openItem = new JMenuItem ("Open File ");

saveItem = new JMenuItem ("Save ");

exitItem = new JMenuItem ("Exit ");

fileMenu.add(openItem);

fileMenu.add(saveItem);

fileMenu.add(exitItem);

menuBar.add(fileMenu);

openItem.addActionListener (this);

saveItem.addActionListener (this);

exitItem.addActionListener (this);

//---------MenuBar------------------------------------//

this.setJMenuBar (menuBar);

this.add(fontLabel);

this.add(fontSizeSpinner);

this.add(fontColorButton);

this.add (fontbox);

this.add(boldTextButton);

this.add(backgroundColorChangeLabel);

this.add (backgroundColorButton);

this.add(modeButton);

this.add(scrollPane);

this.setVisible (true);

}

@Override

public void actionPerformed (ActionEvent e) {

if(e.getSource () == fontColorButton){

JColorChooser colorChooser = new JColorChooser ();

Color color = colorChooser.showDialog(null,"Choose a color",Color.black);

textArea.setForeground (color);

}

if (e.getSource () == fontbox){

textArea.setFont (new Font ((String) fontbox.getSelectedItem (),textArea.getFont ().getStyle (),textArea.getFont ().getSize ()));

}

//code to change background color

if(e.getSource () == backgroundColorButton){

JColorChooser colorChooser = new JColorChooser ();

Color color = colorChooser.showDialog(null,"Choose a color",Color.white);

textArea.setBackground (color);

backgroundColorButton.setText ("Color");

}

//code to get toggle between night mode

if(e.getSource () == modeButton){

if(modeButton.getText () == "Night Mode"){

textArea.setBackground (new Color (160, 162, 162));

textArea.setForeground (Color.white);

this.getContentPane ().setBackground (new Color (0x464648));

modeButton.setText ("Normal Mode");

fontLabel.setForeground (Color.white);

backgroundColorChangeLabel.setForeground (Color.white);

this.getContentPane ().setForeground (Color.white);

}

else{

textArea.setBackground (Color.white);

this.getContentPane ().setBackground (new Color (0xEDEDEE));

textArea.setForeground (Color.black);

fontLabel.setForeground (Color.black);

backgroundColorChangeLabel.setForeground (Color.black);

modeButton.setText ("Night Mode");

}

}

if(e.getSource () == boldTextButton){

if(boldTextButton.getText () == "Bold Text") {

textArea.setFont (new Font (textArea.getFont ().getFamily (), Font.BOLD, textArea.getFont ().getSize ()));

boldTextButton.setText ("Unbold");

}

else{

textArea.setFont (new Font (textArea.getFont ().getFamily (), Font.PLAIN,textArea.getFont ().getSize ()));

boldTextButton.setText ("Bold Text");

}

}

if(e.getSource () == openItem){

JFileChooser fileChooser = new JFileChooser ();

fileChooser.setCurrentDirectory (new File ("."));

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

fileChooser.setFileFilter (filter);

int response = fileChooser.showOpenDialog (null);

if(response == JFileChooser.APPROVE\_OPTION){

File file = new File (fileChooser.getSelectedFile ().getAbsolutePath ());

Scanner fileIn = null;

try{

fileIn = new Scanner (file);

if(file.isFile ()){

while (fileIn.hasNextLine ()){

String line = fileIn.nextLine ()+"\n";

textArea.setText ("");

textArea.append (line);

}

}

}

catch (FileNotFoundException e1){

e1.printStackTrace ();

}

finally {

fileIn.close ();

}

}

}

if(e.getSource () == saveItem){

JFileChooser fileChooser = new JFileChooser ();

fileChooser.setCurrentDirectory (new File ("."));

int response = fileChooser.showSaveDialog (null);

if(response == JFileChooser.APPROVE\_OPTION){

File file;

PrintWriter fileOut = null;

file = new File (fileChooser.getSelectedFile ().getAbsolutePath ());

try {

fileOut = new PrintWriter (file);

fileOut.println (textArea.getText ());

}

catch (FileNotFoundException e1){

e1.printStackTrace ();

}

finally {

fileOut.close ();

}

}

}

if(e.getSource () == exitItem){

System.exit (0);

}

}

public static void main (String[] args) {

new TextEditor ();

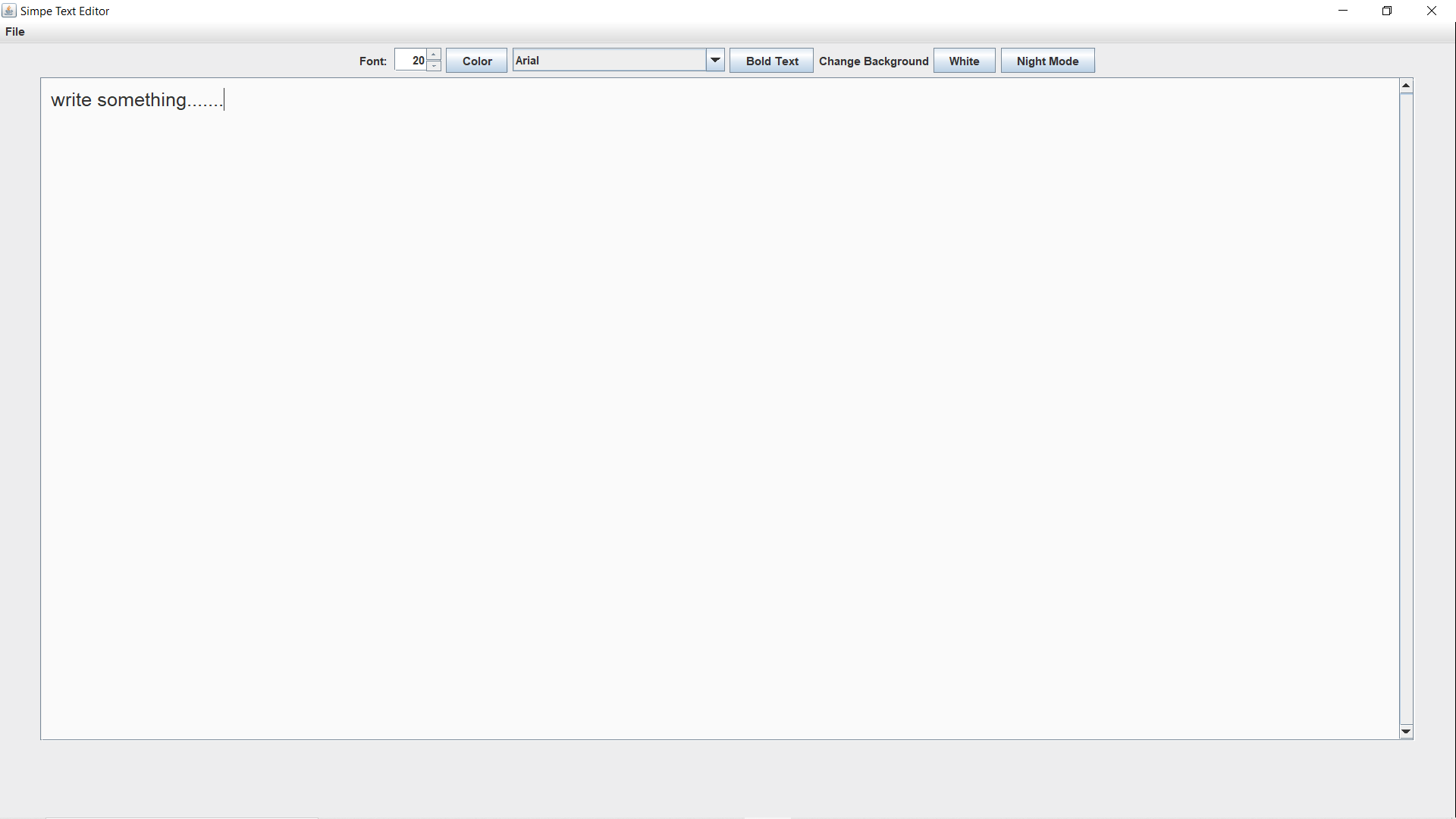
}

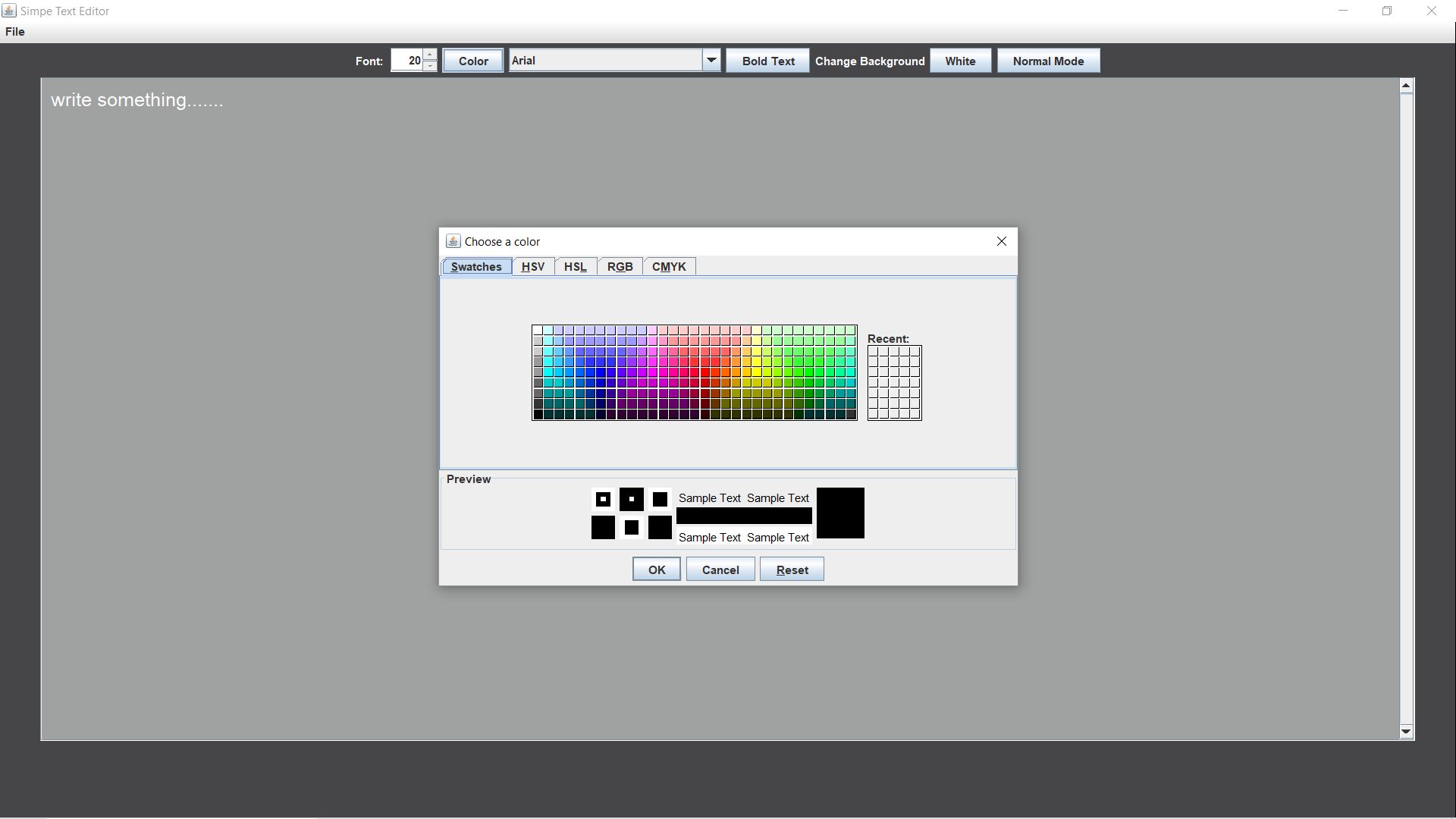
}

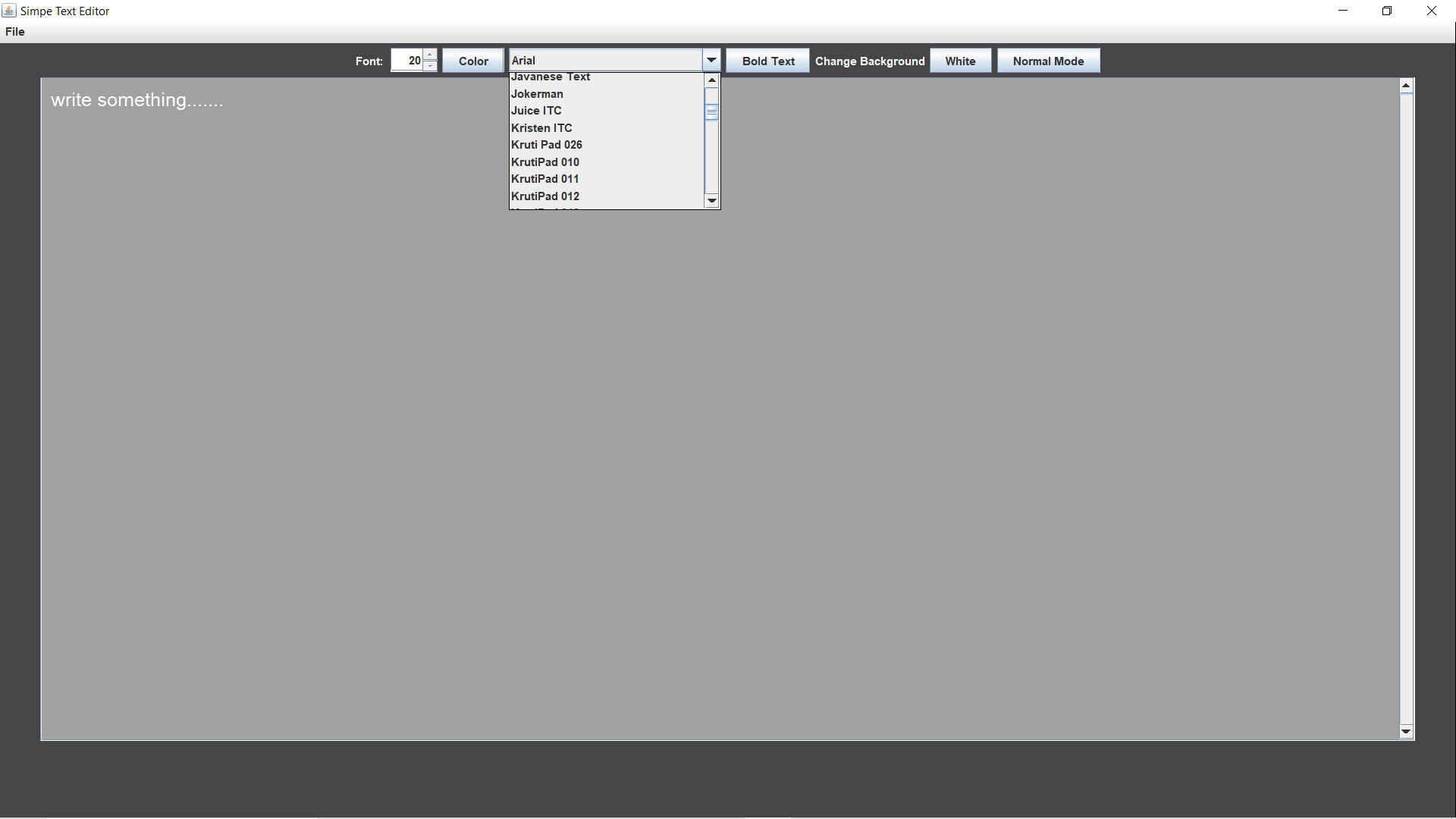
*Testing*

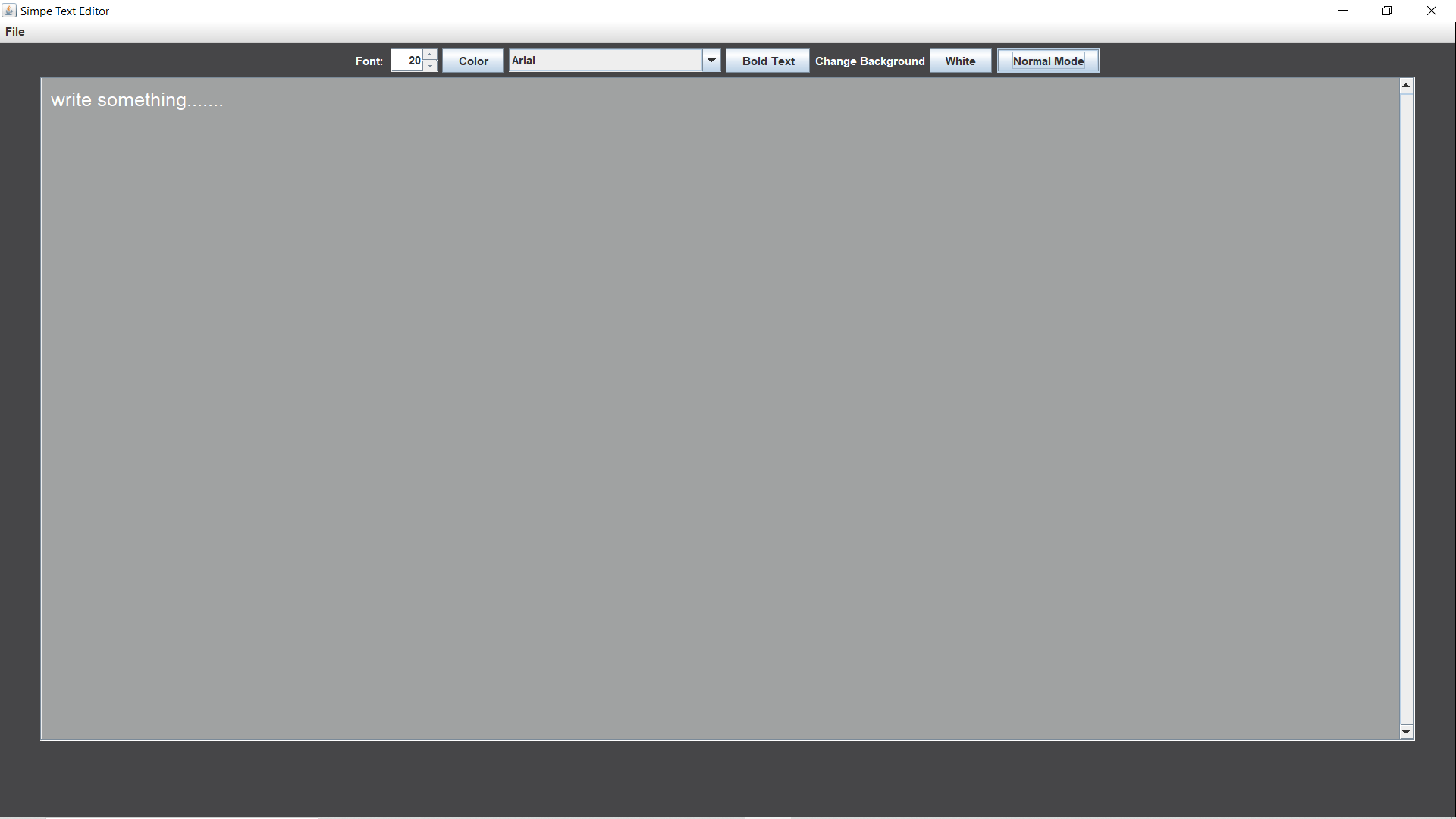
*Testing is the process which is done after the completion of code to check whether the code is correct and working according to the design.*

* *Here are some pictures of the testing and its output result*

**

******

**

**

*Conclusion*

*This project that I undertook was truly a very rewarding experience for me in more than one way. It has given a big thrust to my technical knowledge as prospective Software professional. It has also helped me enhance my skills on the personal front. And I feel extremely satisfied by the fact that I have managed to develop the project of course with equal contribution from my team members. I think I have exploited the opportunity that came my way to the fullest extent by increasing my technical know-how and also gaining the valuable work experience apart from studying the other subjects in our curriculum****.***

Name: *Mohd Ahad*

Roll No: 50

Registration No: 12115116

Contribution in the project:

I have written the code that allows the user to change the font, font size, font color, background color, and style of the text. It also has the functionality to open and save files.

The TextEditor constructor sets up the JFrame by setting the default close operation, title, size, layout, and location. It also sets the background color of the content pane.

The text area is created with default text, font, and background color. The line wrap and wrap style word properties are set to true, and the margin is set to 10 pixels on all sides.

The scroll pane is created with the text area and is set to have a vertical scroll bar always displayed.

Name: *Mohd Raise*

Roll No: 20

Registration No: 12115470

Contribution in the project:

I have created the entire button used in this app such as bold text, change background, night mode and font color.

The actionPerformed method is used to handle user interactions with the GUI. For example, if the user clicks the font color button, a JColorChooser dialog is displayed to allow the user to select a color, which is then applied to the text. If the user selects a font type from the font box, the font of the text is changed to the selected font. If the user clicks the background color button, a JColorChooser dialog is displayed to allow the user to select a background color, which is then applied to the text area. If the user clicks the mode button, the GUI changes between a light and dark mode.

Name: *Md Lucky Ahmad*

Roll No: 31

Registration No: 12115110

Contribution in the project:

I have write a program that toggle night mode and bold text and open save files and save files

Night mode toggle: When the modeButton is clicked, it changes the background and foreground colors of the textArea and the JFrame. It also changes the text on the button to "Normal Mode". If clicked again, it reverts back to the original colors and text on the button changes to "Night Mode".

Open file: When the openItem is clicked, it opens a JFileChooser dialog to select a .txt file to open. It then reads the contents of the file and displays it in the textArea.

Save file: When the saveItem is clicked, it opens a JFileChooser dialog to select the location and name of the file to save. It then saves the contents of the textArea in a .txt file with the selected name.