International institute of Professional Studies, Devi Ahilya University

INDORE, (M.P)

Project Report on

"NOTEPAD IN JAVA"

MTech (IT) - 7th semester

Guided By:

Dr. Shaligram Prajapat

Submitted By:

Adarsh Yagik (IT-2k17-03)

ACKNOWLEDGEMENT

This Project report was completed as a result of support from you. I am grateful to all those people who extended their support and completion of my project.

I wish to express my sincere gratitude to God for his protection, providence, guidance and above all, for sustaining me.

I am greatly indebted to our good supervisor **Dr. Shaligram Prajapat** for his useful and necessary observation, suggestions, contribution and corrections. I would not have been able to achieve anything in this research without your supervision. May God enrich you greatly in every area of life.

Finally, I wish to express our appreciation to my parents for their love and support.

- **Adarsh Yagik**

TABLE OF CONTENTS

1. PREFACE

This project was undertaken at **International institute of Professional Studies, Indore** for the minor project submission in the 7th semester of MTech (IT)-5yrs. The project is named as **NOTEPAD.**

The purpose of this report is to assemble under one cover a sufficient body of knowledge about management and development a successful of Desktop application project. The following quotes outline the basic idea behind this technical report.

This report is about the adaptation of the techniques of project development and reflects the practice and methods of software engineering project this report is intended for:

• Project coordinators —the tutorial presents the state of the practice in software development and management techniques.

• Programmers, analysts, and other computer personnel —the report contains a general description of—and problems in—Desktop application project development, plus a number of methodologies and techniques for managing a software development project.

2. ABSTRACT

I have developed a TEXT EDITOR using Eclipse in Java. It exhibits the basic functionalities which any text editor should have but I have personalized. Eclipse was chosen because any editor requires a good Graphical User Interface and Eclipse provide that.

The functions incorporated needed accurate syntaxes for proper functioning and this was done by Java.

My next challenge was to know all syntaxes and put them in proper places in the source code.

3. AIM OF THE PROJECT

The main aim and reason behind the designing and developing this desktop application, Java primarily based Software project is to provide attractive, user friendly and effective interface to the user over the various operating system.

Java packages, oops concept, file handling used to develop this desktop application. Where all the users can do their basic stuffs like making notes, writings journals, managing docs. Users will have all the options and features in this application like creating new file, saving the existing file, printing the file, copy, paste, cut, selectall the text field.

4. INTRODUCTION

The project is based on **“Text Editor".** This software firm deals in developing software for its clients.

Text Editor: - A text editor is a type of program used for editing plain text files. A plain text file is represented and edited by showing all the characters as they are present in the file. The only characters usable for 'mark-up' are the control characters of the used character set; in practice this is newline, tab and form feed. The most commonly used character set is, ASCII especially recently, as plain text files are more often being used for programming and configuration, and less frequently for documentation (e.g., detailed instructions, user guides) than in the past.

This text editor developed at JAVA platform is a replica of the word editors we all are familiar with and which we use quite often on a daily basis. The only difference being that, this editor has been created using JAVA for the front-end interface. The text edited in the editor is stored in the desired location and for this operation, the no-fuss FILE storage system

4.1 INTRODUCTION TO ENVIRONMENT

This Package has been developed in **Java**, which is based on Object Oriented Methodology. There are several packages resided in Java and to develop this system, mainly awt and swing packages of Java are used. Various features of Java made it a first choice of the programmers. Java is a platform independent language, which can be run under any kind of environment.

The main features involve,

1.Simple: - Java is a language which is based on Object Oriented Methodology, so it is very easy to learn and can be used effectively.

2.Robust: - Java Programs are said to be robust because they will take care of memory management and will never crash under any circumstances.

3.Secure: - Even though Java is developed using Object Oriented Principles, it eliminated the Pointers Concept. So, it is not possible to access memory directly, that's why Java is said to be Secure and is applicable for Internet, for that Applet is designed which can be understandable by the browsers.

4.Portable: - Java Programs are Portable that those can be run under any kind of environment irrespective of the hardware used. This is known as platform independent.

5.Compiled & Interpreted: - Unlike remaining Programming languages Java code is both Compiled and Interpreted. The output after compilation is 'Byte Code' which is interpreted to produce output. This Byte Code is a new evolution, which makes Java a Platform Independent Language

4.2 FEATURES OF NOTEPAD

Our text editor provides basic features of an editor.

- New File –User can save the existing files and can create new files.
- File open – User can either open the files already existing in the system or open a new blank file.
- Files save - one can save the file in any desired format like-.txt, .doc, .java etc. The file is stored in the location specified by the user.
- File print –User can easily print their files.
- Cut-Copy-Paste – This editor also lets the user cut-copy-paste the edited text.
- SelectAll – User can select all the text written in the text field from here.
- About the Notepad- Here user will get basic information of the notepad.

4.3 PROJECT OVERVIEW

Project Name	**:**	Notepad
Institute	**:**	International institute of Professional Studies
Front End	**:**	Java Swings Packages

Back End	**:**	Basic Java
Project Type	**:**	Desktop Application

5. 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 Requirements Analysis
3. Systems Analysis and Design
4. Code Generation
5. Testing
6. Maintenance

5.1. System/Information Engineering and Modeling

As software is always of a large system (or business), 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 resources. 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.

5.1.1. Basic Hardware Configuration Requirement

- Console: - Mouse, Monitor, Keyboard.
- Processor: - Intel Celeron or higher,
- RAM: - 1Gb RAM
- ROM: - 64 GB Hard Disk

5.2. 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.

5.2.1. Software Requirements

OPERATING SYSTEM: Any Operating System

COMPILER USED: Java Installed Platform.

IDE USED: Eclipse

Front End: Java Swings Package

Language: English

5.3. 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.

5.3.1. Software Used: -

Some of the features of Java that are used in this system include, Multithreaded programming that allows running 2 or more processes concurrently, File Handling for saving, opening and printing file.

5.3.2. Packages Used: -

1. The default package to build any program in Java is **lang** that is used in developing this system
2. The API to build Graphical User Interface in Java is provided using the packages **awt** and **swing**. This system uses both the packages to provide GUI based forms.
3. The streams that are used to write and read the data over system are also utilized in this system, which are in the package of **io**.

5.4. Code generation

The design must be translated into a machine-readable form. The code generation step performs this task. If the design is performed in a detailed manner, code generation can be accomplished without much complication. Programming tools like Compilers, Interpreters, and Debuggers are used to generate the code. Different high level programming languages like C, C++, Pascal, and Java are used for coding. With respect to the type of application, the right programming language is chosen.

5.4.1. IMPLEMENTING TEXT EDITOR APPLICATION

Notepad application includes:

- Importing the inbuilt Java libraries.
- Declaring the variables.
- Designing the Notepad interface.
- Adding listeners for tracking various
- Events of the Notepad.
- Handling of text editing events.

5.5. Testing

Once the code is generated, the software program testing begins. Different testing methodologies are available to unravel the bugs that were committed during the previous phases. Different testing tools and methodologies are already available.

Some companies build their own testing tools that are tailor made for their own development operations.

5.6. Maintenance

Software will definitely undergo change once it is delivered to the customer. There are many reasons for the change. Change could happen because of some unexpected input values into the system. In addition, the changes in the system could directly affect the software operations. The software should be developed to accommodate changes that could happen during the post implementation period.

5.6.1. Updates Required

- It should have the ability to Search and Replace Text.
- A file should be locked with password protection.
- Colorful Text would also be entered in notepad.
- Size Limitation Should be removed.
- Theme option should be there for attractive interface.

6. DATA FLOW DIAGRAM

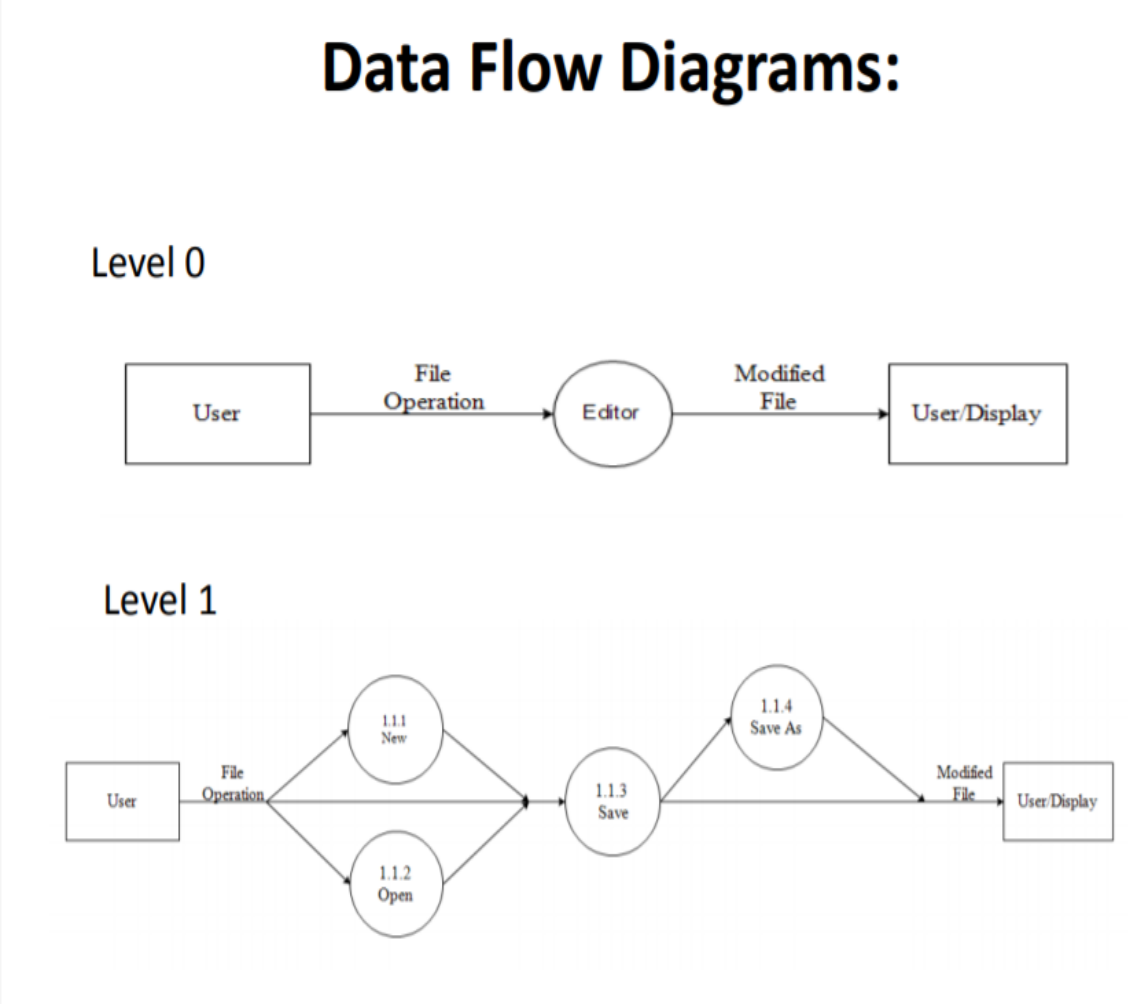


Figure: Data Flow Diagram of Notepad

7. ARCHITECTURE OF NOTEPAD APPLICATION

The main menu bar of the Notepad application consist of three pull down menus:

8. 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. 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.

9. REFRENCES

1. Java Packages: https://www.w3schools.com/java/java_packages.asp
2. Creating a GUI using swing packages in java: https://www.edureka.co/blog/java-swing/
3. Use of awt in java: https://beginnersbook.com/2015/06/java-awt-tutorial/
4. File handling in Java using FileWriter and FileReader: https://www.geeksforgeeks.org/file-handling-java-using-filewriter-filereader/

10. SOURCE CODE:

```
package Notepad;
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import javax.swing.*;
import javax.swing.filechooser.*;

public class Notepad extends JFrame implements ActionListener
{

JTextArea area;
String text;
Notepad(){

setBounds(0,0,1920,1080);
```

```
JMenuBar menubar = new JMenuBar();

JMenu file = new JMenu("File");

JMenuItem newdoc = new JMenuItem("New");
newdoc.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK
_N,ActionEvent.CTRL_MASK));
newdoc.addActionListener(this);

JMenuItem open = new JMenuItem("Open");
open.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_O,
ActionEvent.CTRL_MASK));
open.addActionListener(this);

JMenuItem save = new JMenuItem("Save");
save.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_S,
ActionEvent.CTRL_MASK));
save.addActionListener(this);

JMenuItem print = new JMenuItem("Print");
print.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_P,
ActionEvent.CTRL_MASK));
print.addActionListener(this);
```

```
JMenuItem exit = new JMenuItem("Exit");
exit.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_ESCAPE,0));
exit.addActionListener(this);

file.add(newdoc);
file.add(open);
file.add(save);
file.add(print);
file.add(exit);

JMenu edit = new JMenu("Edit");

JMenuItem copy = new JMenuItem("Copy");
copy.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_C, ActionEvent.CTRL_MASK));
copy.addActionListener(this);

JMenuItem paste = new JMenuItem("Paste");
paste.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_V, ActionEvent.CTRL_MASK));
paste.addActionListener(this);

JMenuItem cut = new JMenuItem("Cut");
```

```
cut.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_X,ActionEvent.CTRL_MASK));
cut.addActionListener(this);

JMenuItem selectall = new JMenuItem("SelectAll");
selectall.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_A,ActionEvent.CTRL_MASK));
selectall.addActionListener(this);

edit.add(copy);
edit.add(paste);
edit.add(cut);
edit.add(selectall);

JMenu help = new JMenu("Help");
JMenuItem about = new JMenuItem("About The Notepad");
about.addActionListener(this);

help.add(about);

area = new JTextArea();
JScrollPane pane = new JScrollPane(area);
add(pane,BorderLayout.CENTER);
area.setFont(new Font("Verdana", Font.PLAIN, 20));
```

```
area.setLineWrap(true);

menubar.add(file);
menubar.add(edit);
menubar.add(help);

setJMenuBar(menubar);
}

public void actionPerformed(ActionEvent ae)
{
if (ae.getActionCommand().equals("New"))
{
area.setText("");
}
else if (ae.getActionCommand().equals("Open"))
{
JFileChooser chooser = new JFileChooser();
chooser.setAcceptAllFileFilterUsed(false);
FileNameExtensionFilter restrict = new
FileNameExtensionFilter("Only .txt file", "txt");
chooser.addChoosableFileFilter(restrict);
int action = chooser.showOpenDialog(this);
```

```
if (action != chooser.APPROVE_OPTION)
{
return;
}

File file =chooser.getSelectedFile();
try
{
BufferedReader reader = new BufferedReader(new
FileReader(file));
area.read(reader, null);

}
catch (Exception e)
{

}
}
else if (ae.getActionCommand().equals("Save"))
{
JFileChooser saveas = new JFileChooser();
saveas.setApproveButtonText("Save");
int action = saveas.showOpenDialog(this);
```

```
if (action != saveas.APPROVE_OPTION)
{
return;
}
File filename = new File(saveas.getSelectedFile() + ".txt");
BufferedWriter outfile = null;
try
{
outfile = new BufferedWriter(new FileWriter(filename));
area.write(outfile);
}
catch (Exception e)
{

}

}
else if (ae.getActionCommand().equals("Print"))
{
try
{
area.print();
```

```
}
catch (Exception e) {}
}
else if (ae.getActionCommand().equals("Exit"))
{
System.exit(0);
}


else if(ae.getActionCommand().equals("Copy"))
{
text = area.getSelectedText();
}
else if(ae.getActionCommand().equals("Paste"))
{
area.insert(text, area.getCaretPosition());
}
else if(ae.getActionCommand().equals("Cut"))
{
text = area.getSelectedText();
area.replaceRange("",area.getSelectionStart(),
area.getSelectionEnd());
}
```

```
else if(ae.getActionCommand().equals("SelectAll"))
{
area.selectAll();
}
else if (ae.getActionCommand().equals("About The Notepad"))
{
new About().setVisible(true);
}

}

public static void main(String[] args)
{
new Notepad().setVisible(true);
}

}
```

//ABOUT CLASS

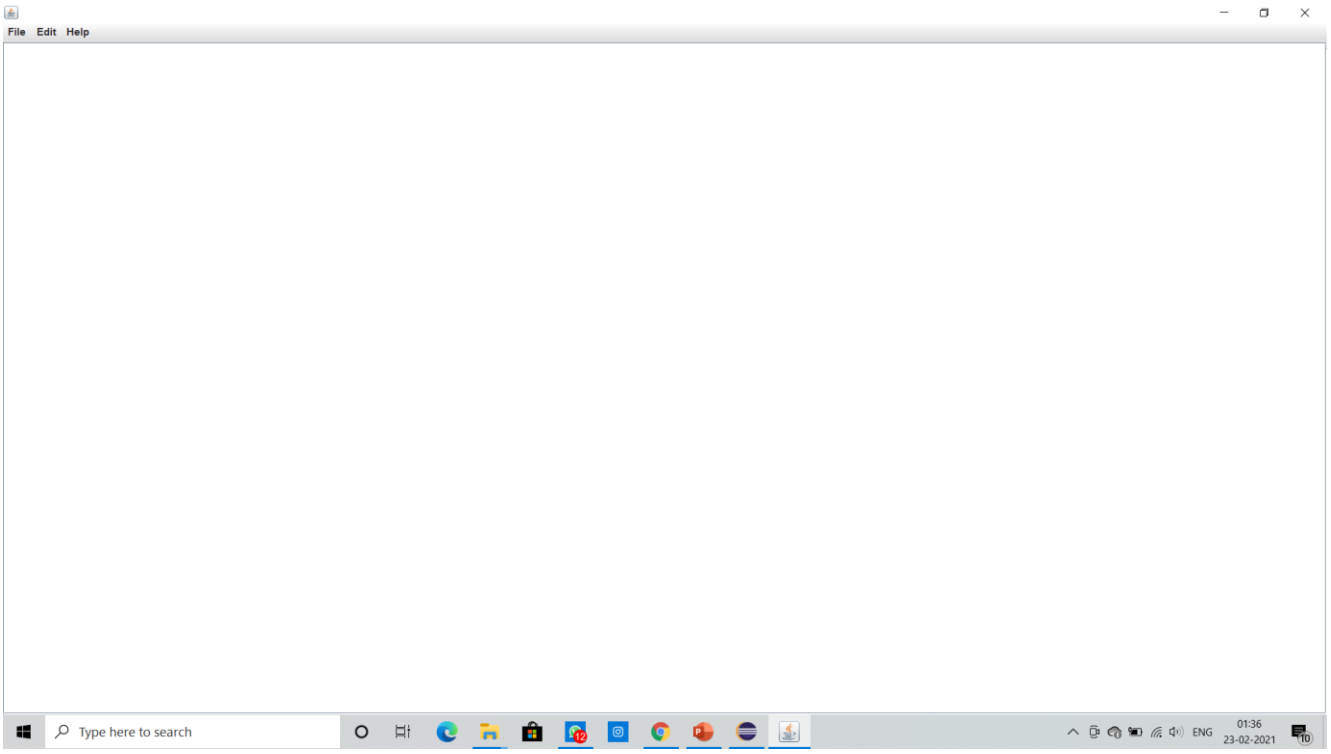
```
package Notepad;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
```

```
public class About extends JFrame implements ActionListener {

JButton b1;
About(){
setBounds(500,150,600,500);
setLayout(null);

ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("Notepad/icons/Windows.png"));
Image i2 = i1.getImage().getScaledInstance(400, 80,
Image.SCALE_DEFAULT);
ImageIcon i3 = new ImageIcon(i2);
JLabel l1 = new JLabel(i3);
l1.setBounds(100, 40, 400, 80);
add(l1);

ImageIcon i4 = new
ImageIcon(ClassLoader.getSystemResource("Notepad/icons/Notebook.png"));
Image i5 = i4.getImage().getScaledInstance(70, 80,
Image.SCALE_DEFAULT);
ImageIcon i6 = new ImageIcon(i5);
JLabel l2 = new JLabel(i6);
l2.setBounds(40, 150, 70, 70);
add(l2);

JLabel l3 = new JLabel("<html><br>Notepad is word
prosessing program<br><br>which allows change of text in
```

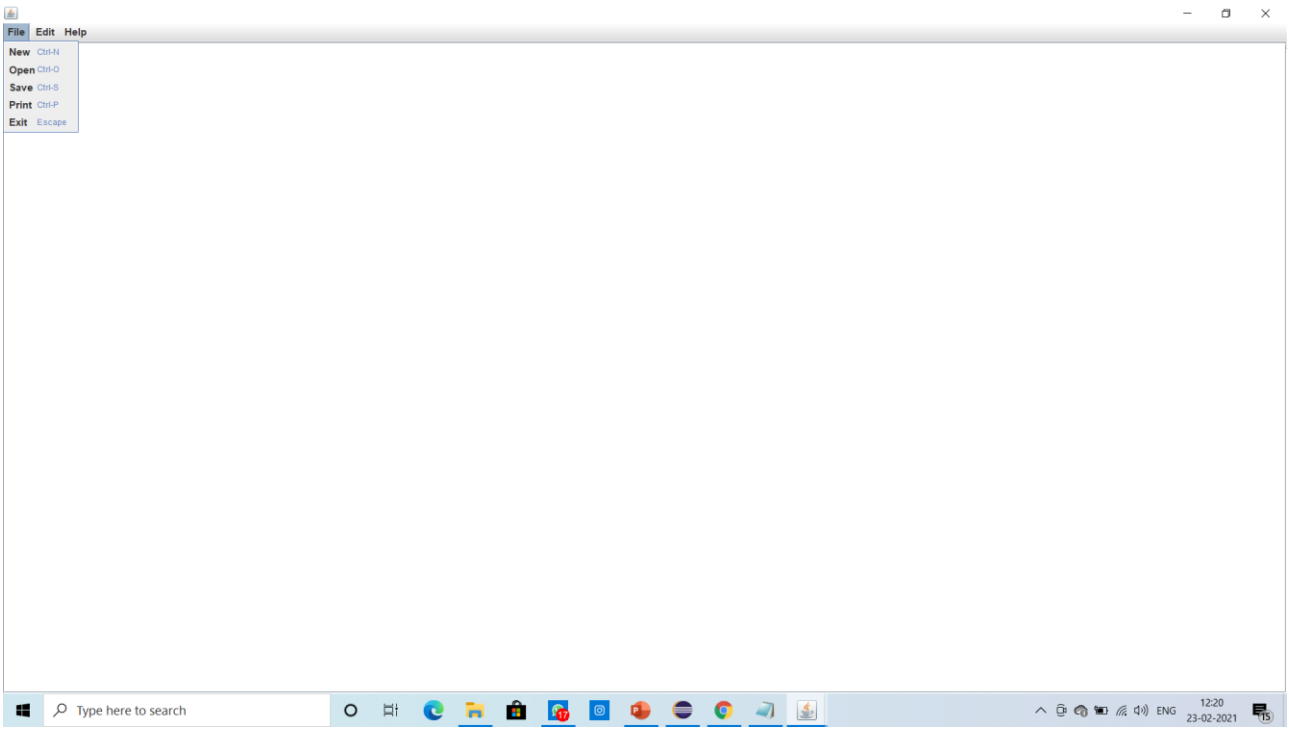
```
computer files<br><br>Notepad is simple text editor for basic
text editing program<br><br>which enables user to create
documents</html>");
l3.setBounds(150, 130, 500, 180);
l3.setFont(new Font("SAN SERIF", Font.PLAIN,16));
add(l3);

b1 = new JButton("OK");
b1.setBounds(250,400, 80, 25);
add(b1);
b1.addActionListener(this);
}

public void actionPerformed(ActionEvent ae)
{
this.setVisible(false);

}

public static void main (String args[]) {

new About().setVisible(true);
}
}
```

10. OUTPUT:

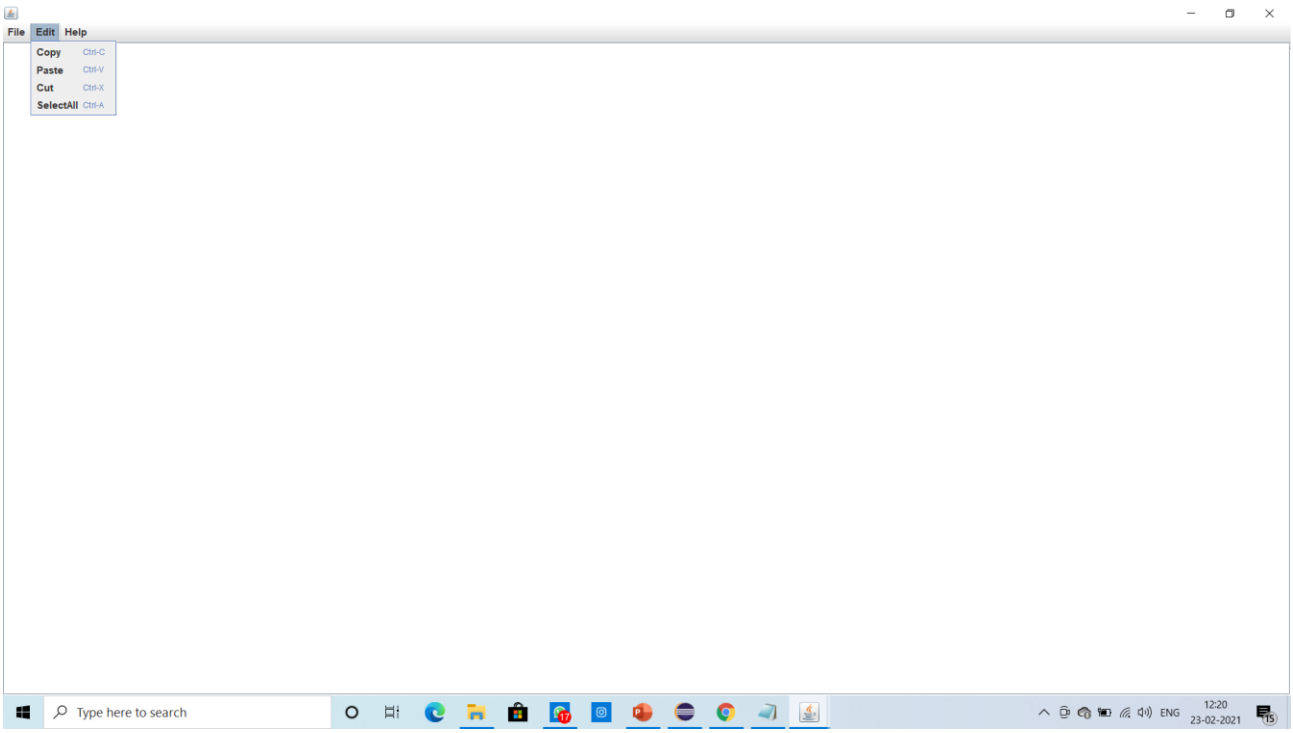
Notepad Application:



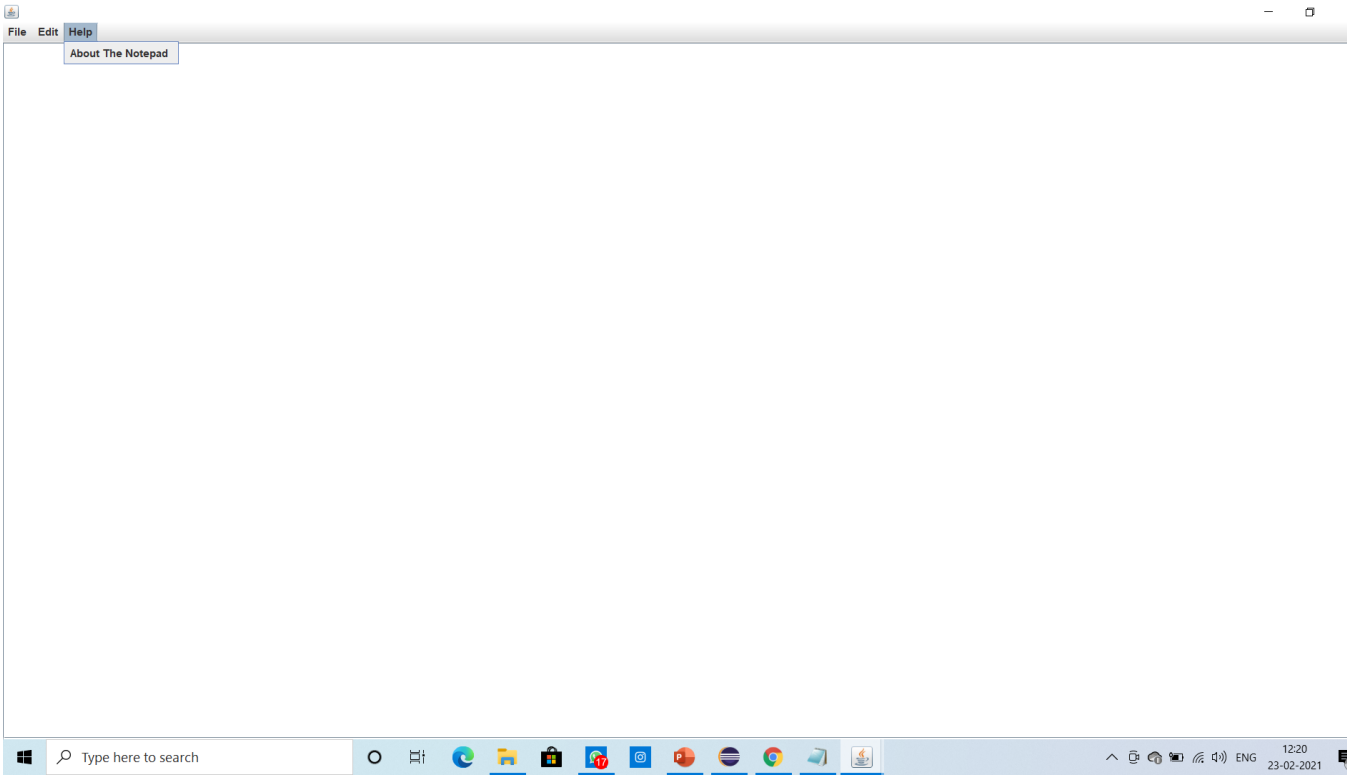
File Menu:


File Edit Help
New Ctrl-N
Open Ctrl-O
Save Ctrl-S
Print Ctrl-P
Exit Escape
Type here to search
ENG
12:20
23-02-2021

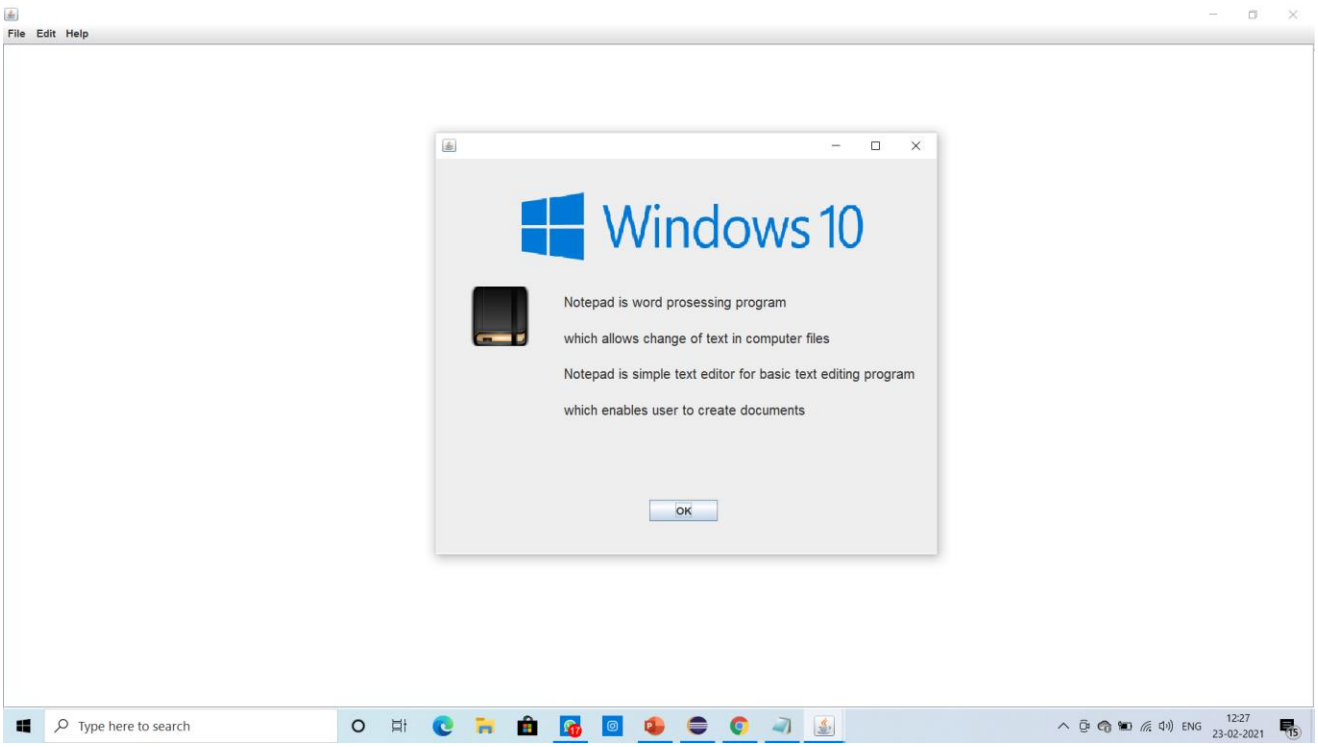
Edit Menu:


File Edit Help
Copy Ctrl-C
Paste Ctrl-V
Cut Ctrl-X
SelectAll Ctrl-A
Type here to search
ENG
12:20
23-02-2021

AboutMenu:



About the Notepad:


File Edit Help
Windows 10
Notepad is word prosessing program
which allows change of text in computer files
Notepad is simple text editor for basic text editing program
which enables user to create documents
OK
Type here to search
ENG
12:27
23-02-2021