

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**  
**BELAGAVI, KARNATAKA**



*A Mini Project Report*

*(Sixth semester)*

*on*

**“BASIC TEXT EDITOR USING JAVA”**

Submitted in the partial fulfillment for the requirements for the FS Lab with Mini Project  
(18ISL68)

**BACHELOR OF ENGINEERING**

*in*

**INFORMATION SCIENCE AND ENGINEERING**

*by*

**Mr. RAGHAVENDRA K M**

USN: 1BY18IS093

**Ms. SANJANA GAJANANA SHETTY**

USN: 1BY18IS104

Under the guidance of

**Mrs. Shanthi D L**

Assistant Professor

Dept of ISE, BMSIT



**BMS INSTITUTE OF TECHNOLOGY & MANAGEMENT**  
**YELAHANKA, BENGALURU-560064**  
**DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**



**2020-2021**

# **BMS INSTITUTE OF TECHNOLOGY & MANAGEMENT**

**YELAHANKA, BENGALURU-560064**

## **DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**



### **CERTIFICATE**

This is to certify that the Project work entitled “**BASIC TEXT EDITOR USING JAVA**” is a bonafide work carried out by **Mr. Raghavendra K M (1BY18IS093)**, **Ms. Sanjana Gajanana Shetty (1BY18IS104)** in partial fulfillment of File Structures Lab with Mini Project (18ISL68) for the award of **Bachelor of Engineering Degree in Information Science and Engineering** of the Visvesvaraya Technological University, Belagavi during the year 2020-21. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in this report. The project report has been approved as it satisfies the academic requirements in respect of Mini Project work for the B.E Degree.

---

#### **Signature of the Guide**

Mrs. Shanthi D L  
Assistant Professor  
Department of ISE

---

#### **Signature of the HOD**

Dr. Pushpa S.K  
Professor and Head  
Department of ISE

#### **EXTERNAL EXAMINERS**

Name of the Examiners

- 1.
- 2.

Signature with Date

## ACKNOWLEDGEMENT

We are happy to present this mini project after completing it successfully. This mini project would not have been possible without the guidance, assistance and suggestions of many individuals. We would like to express our deep sense of gratitude and indebtedness to each and every one who has helped us make this mini project a success.

We heartily thank our **Principal, Dr. Mohan Babu G.N, BMS Institute of Technology & Management** for his constant encouragement and inspiration in taking up this mini project.

We heartily thank our **Head of Department Dr. Pushpa S.K, Dept. of Information Science and Engineering, B M S Institute of Technology& Management** for his constant encouragement and inspiration in taking up this mini project.

We gratefully thank our Project guide, **Mrs. Shanthi D L, Asst. Professor, Dept. of Information Science and Engineering**, for her encouragement and advice throughout the course of the mini project.

Special thanks to all the staff members of the Information Science Department for their help and kind cooperation.

We also thank our parents and friends for their unconditional love and encouragement and support given to us in order to finish this precious work.

Last but not the least we would like to thank God for giving us the strength and motivation through the course of this Project.

By,

**Raghavendra K M**

**Sanjana Gajanana Shetty**

## **ABSTRACT**

A useful text editor is an essential component of any personal computer. If the task is to create a simple todo list or else to create a html page, either of the times we make use of the text editor in our day-to-day life. Not just that, but also to make notes, compose documents, and record vital pieces of information. We depend on our text editor. In a broad sense, the application provides us with the ability to create, edit, and save files containing text. Text editors are the most commonly used software for writing documentation, plain texts and project source codes. Notepad, Wordpad, etc. are some of the popular text editor tools with various features and options. During project design, development and implementation, text editors play an important role.

A text editor is a computer program that lets a user open, modify, save, and usually read text. And also, the advantage here is, it doesn't make the fans of your systems run harder, because it's a lightweight software. Therefore, the software is capable of running in any system.

# TABLE OF CONTENTS

Chapter 1 Introduction	1
1.1 Outline	1
1.2 Motivation and Scope	1
1.3 Problem Statement	2
1.4 Limitations	2
Chapter 2 Requirements Specification	3
2.1 Functional Requirements	3
2.2 Non-Functional Requirements	4
2.3 Domain Constraints	4
Chapter 3 Requirements and System Analysis	5
3.1 Overall Process of the Project	5
3.2 Components/Subsystem Design	6
Chapter 4 System Design	8
4.1 UI Logic Interface/Interaction Details	8
Chapter 5 Implementation	10
5.1 Description of Frameworks Used	10
5.2 Description of Integrated Development Environment	11
Chapter 6 Testing	13
6.1 Component Tests	13
6.2 System Tests	15
Chapter 7 Interpretation of Results	16
Conclusion	20
References	21

## Chapter 1

# INTRODUCTION

### 1.1 Outline

An Interactive text editor has become an important part of almost any computing environment. Text editor acts as a primary interface to the computer for all types of “knowledge workers” as they compose, organize, study, and manipulate computer-based information. A text editor allows you to edit a text file.

Text editors differ from word processors, such as Microsoft Word or WordPerfect, in that they do not add additional formatting information to documents. You might write a paper in Word, because it contains tools to change fonts, margins, and layout, but Word by default puts that formatting and layout information directly into the file, which will confuse the compiler. If you open a .doc file in a text editor, you will notice that most of the file is formatting codes. Text editors, however, do not add formatting codes, which makes it easier to compile your code. Text editors have a feature set different from that of a traditional word processing program. For example, most won't let you include pictures, or include tables, or double-space your writing.

The features of text editors vary from implementation to implementation, but there are several kinds of features that most editors have. We can edit files of almost any format like txt, html etc. Notepad offers only the most basic functions, such as finding text. Easy to use implementation of Shortcut Keys for Basic Editing Functions. Why Notepad in Java? As we all know, Java is a platform independent Language. It can Run on any Operating System which has a JVM installed.

The Simplified and User-Friendly Interface of Notepad makes it most reliable for any kind of basic Text Editing. Basic Features like shortcuts for common tasks like new, open and save files. Notepad can edit traditional 8-bit text files. When clipboard data with multiple formats is pasted into Notepad, the program will only accept text in the txt format. Time/Date feature will automatically insert the current time and date.

## 1.2 Motivation and Scope

The first thing we need to do is to write some code. This obviously involves tapping away on the computer keyboard, but we are also dependent on software to record and manage our keystrokes in an effective manner. An important feature of the computer code that we will write is that it is just plain text. In the absence of everything else, a text editor is better than using a word processor. Text editors do a much better job. When we write computer code, a good choice of text editor can make us much more accurate and efficient. Text editors are the most commonly used software for writing documentation, plain texts and project source codes. Notepad, Wordpad, etc. are some of the popular text editor tools with various features and options. During project design, development and implementation, text editors play an important role.

## 1.3 Problem Statement

Creating a text editor where it performs various tasks like creating new files, opening existing files, saving files, changing font size as well as background color, status bar and word wrap, Copy, Cut, Paste and delete options. After opening the text editor, we can enter data or type using any keyboard. Editing of existing documents can also be done by opening the file.

## 1.4 Limitations

- It doesn't have the ability to Search.
- Font style is not available.
- file should be locked with password protection.
- Insert words
- Find and replace text within the document

## Chapter 2

# REQUIREMENTS SPECIFICATION

### 2.1 Functional Requirements

#### ❖ Creating a File

This requires the user to create a new file handle, use this file handle to enter its contents into a buffer and save the buffer contents to memory when editing it.

#### ❖ Opening an Existing File

Here the user must open a valid file existing on disk and open this file by creating a file handle copy file contents to main memory edit or delete its contents and use the same file handle contents and write the contents to disk.

#### ❖ Saving a File

When a user opens a file, through the help of a file handle the same file handle must successfully save the file contents on disk to the location specified by the user.

#### ❖ Writing a File

Once a user opens a file and decides that he has edited enough the program must save each and every character stored in the file buffer onto disk without faults of any sort.

#### ❖ Modify File Contents

If the user chooses to edit an existing file it should be opened and its content should be presented to the user as it is in the file it should also provide the user the choice either to overwrite the same file or save contents to a new file.



## **2.2 Non-Functional Requirements**

### **Performance**

Performance of Quark should always vary between a few hundred milliseconds. Time taken to create files, open existing files, flush contents of the buffer onto disk when user hits save, find keywords within the file, replacing keywords within the file should be minimal.

### **Reliability**

Quark shall always provide fast and flexible file editing functionality no matter what the environment. Quark shall be robust enough to have a high degree of fault tolerance. For example, if a user tries to open an invalid file it should display a proper error message. It shall be able to recover from hardware failures, power failures and other natural catastrophes and rollback the files to their most recent valid state.

### **Usability**

Quark shall provide an easy-to-use graphical interface similar to other existing text file editors so that the users do not have to learn a new style of interaction.

Any notification or error messages generated by Quark shall be clear, succinct, polite and free of jargon.

### **Integrity**

The system must be programmed properly to prevent exploitation through buffer overflows etc. The system should be secure and could use encryption to protect the files.

Users need to be authenticated before having access to any personal data.

### **Interoperability**

Quark shall minimize the effort required to couple it to another system, such as an Integrated Development Environment.

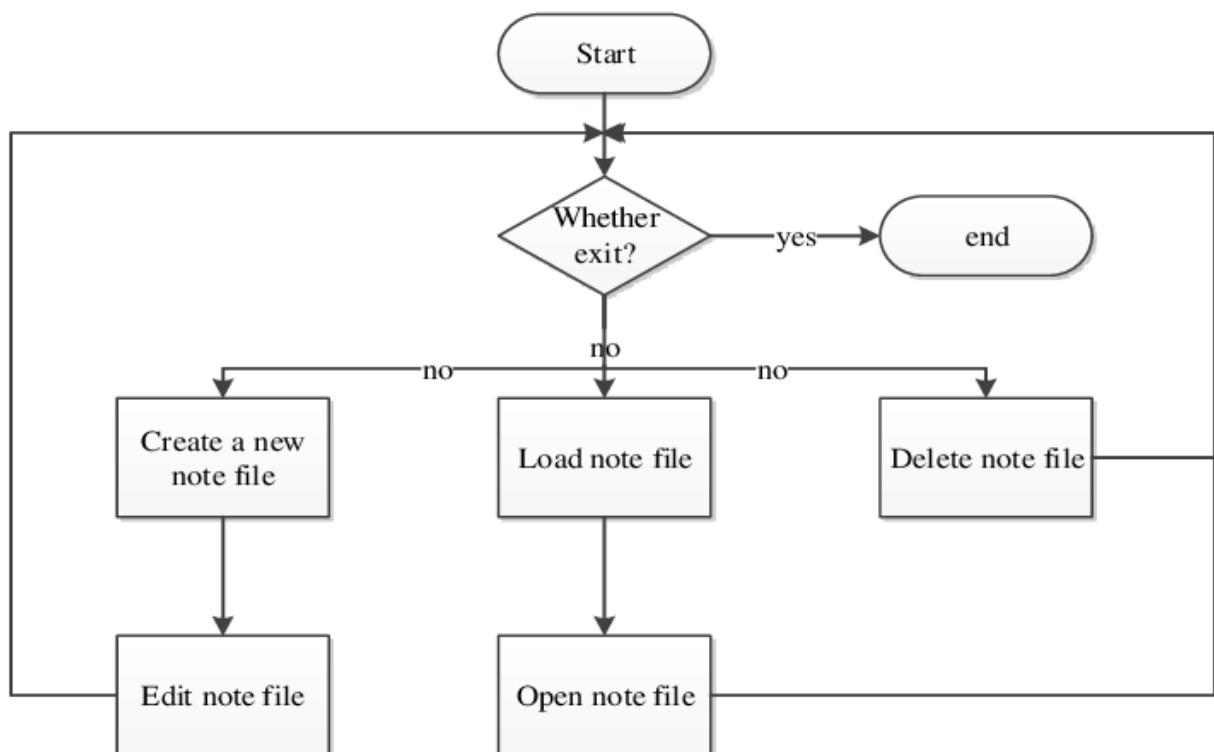
## 2.3 Domain Constraints

- Hardware limitations: There must be a 64 MB on board memory
- Control functions: The software must be very user-friendly and display appropriate error messages.
- Dependencies: Requires JDK 8, Java SE and JFeonix Library.
- Parallel operations: It must support many file operations simultaneously.
- Safety/security considerations: The application must be exited always normally.

## Chapter 3

# SYSTEM/REQUIREMENTS ANALYSIS

### 3.1 Overall System Description



The overall description of the system is as follows:

The user is first presented with the Main Panel. Here the user has a set of three options.

**Create file:** - In this Module User can create a file and write the content to the file. Next, he can perform basic file operations such as changing the content of the file, find and replace any selected word from the file and save the file.

**Open Existing file:** - In this Module the user opens the Existing file from the computer and can edit it.

**Exit:** - If the user wants to close the application, he can do so using the Exit Button.

## **3.2 Components/Subsystem Design**

### **Create File Module**

In the Create file module the user creates and writes the content into a new file. Next, he is provided with two options, Save and Exit. After successful file operation, he can save the file by any name. Lastly, he can exit the window and discard contents by clicking the Exit button.

### **Open File Module**

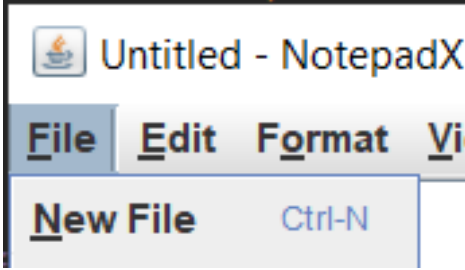
This module allows the user to open the Existing file from the Computer. Next, he is provided with two options, Save and Exit. After successful file operation, he can save the file by any name. Lastly, he can exit the window and discard contents by clicking the Exit button.

## Chapter 4


# SYSTEM DESIGN

### 4.1 UI Logic Interface/Interaction Details

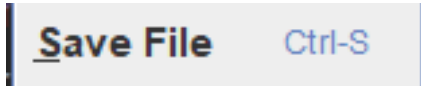
*Fig 4.1.1 JMenuItem New File*

<i>Fig 4.1.1 JMenuItem New File</i>	
	id: fileNew onAction: fileHandler.newFile(); Controller Class: Notepad.java

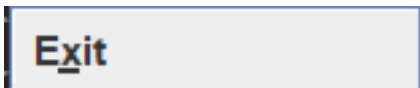
*Fig 4.1.2 JMenuItem Open File*

<i>Fig 4.1.2 JMenuItem Open File</i>	
	Fx:id: fileOpen onAction: fileHandler.openFile(); Controller Class: Notepad.java

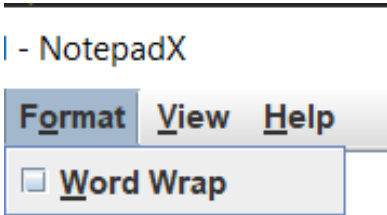
*Fig 4.1.3 JMenuItem Save File*

<i>Fig 4.1.3 JMenuItem Save File</i>	
	Fx:id: fileSave onAction: fileHandler.saveThisFile(); Controller Class: Notepad.java

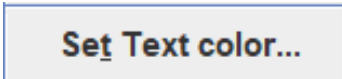
*Fig 4.1.2 JMenuItem Exit File*

<i>Fig 4.1.2 JMenuItem Exit File</i>	
	Fx:id: fileExit onAction: System.exit(0) Controller Class: Notepad.java

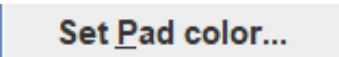
*Fig 4.1.2 JMenuItem Word Wrap*

	<p>Fx:id: formatWordWrap onAction: setLinkWrap Controller Class: Notepad.java</p>
---	---

*Fig 4.1.2 JMenuItem Set Text Color*

	<p>Fx:id: formatForeground onAction: showForegroundColorDialog Controller Class: Notepad.java</p>
---	---

*Fig 4.1.2 JMenuItem Set Pad Color*

	<p>Fx:id: formatBackground onAction: showBackgroundColorDialog Controller Class: Notepad.java</p>
---	---

## Chapter 5

# IMPLEMENTATION

### 5.1 Description of Frameworks Used

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 a subsidiary of Oracle) in 1995. James Gosling is known as the father of Java. Before Java, its name was Oak. Since Oak was already a registered company, James Gosling and his team changed the Oak name to Java.



Platform: Any hardware or software environment in which a program runs, is known as a platform. Since Java has a runtime environment (JRE) and API, it is called a platform.

#### Java Swing Package:

It is a part of Java Foundation Classes (JFC) that is *used to create window-based applications*. It is built on the top of the AWT (Abstract Windowing Toolkit) API and entirely written in java.

Unlike AWT, Java Swing provides platform-independent and lightweight components.

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

### 5.2 Description of Integrated Development Environment

IntelliJ IDEA is an integrated development environment (IDE) written in Java for developing computer software. It is developed by JetBrains (formerly known as IntelliJ), and is available as an Apache 2 Licensed community edition, and in a proprietary commercial edition. Both can be used for commercial development.



## Chapter 6

# TESTING

### 6.1 Component Testing

#### File Menu Module

Table 6.1 File Menu Module Test

TEST UNIT	TEST CASE	RESULT
New Menu item	On click event on the New File button	System opens up a fresh new editor for writing text documents
Open Menu item	On click event on the Open File button	System opens up a window for opening previously saved text documents
Save Menu item	On click event on the Save File button	System opens up a window for saving the current unsaved text document
Exit Menu item	On click event on the Exit button	The system will be closed



## Edit Menu Module

Table 6.2 Edit Menu Module Test

TEST UNIT	TEST CASE	RESULT
Cut Menu item	On click of cut button	The system will copy the contents of the selected text into clipboard and removes the selected text
Copy menu item	On click of Copy button	The system will copy the contents of the selected text into clipboard and removes the selected text
Paste menu item	On click of Copy button	The system will paste the contents present in the clipboard

## Format Menu Module

Table 6.3 Format Menu Module Test

TEST UNIT	TEST CASE	RESULT
Word wrap menu item	On click of Word Wrap button	The system will force all text present in the JTextArea to be confined within the defined margins.
Set Text Color menu item	On click of Set Text Color button	The system will change the color of the text present in the JTextArea.
Set Pad Color menu item	On click of Set Pad Color button	The system will change the background of the JTextArea.

## View book module

Table 6.4 View Module Test

TEST UNIT	TEST CASE	RESULT
Status Bar Menu Item	On check of the Status bar JCheckBox Button	The system will display the Status Bar (current line number and the column number according to the location of the cursor)
Status Bar Menu Item	On uncheck of the Status bar JCheckBox Button	The system will not display Status bar

## Help Module

Table 6.5 Help Module Test

TEST UNIT	TEST CASE	RESULT
About Menu item	On click of the About button	The system will open up a window showing details of the Project and Developers

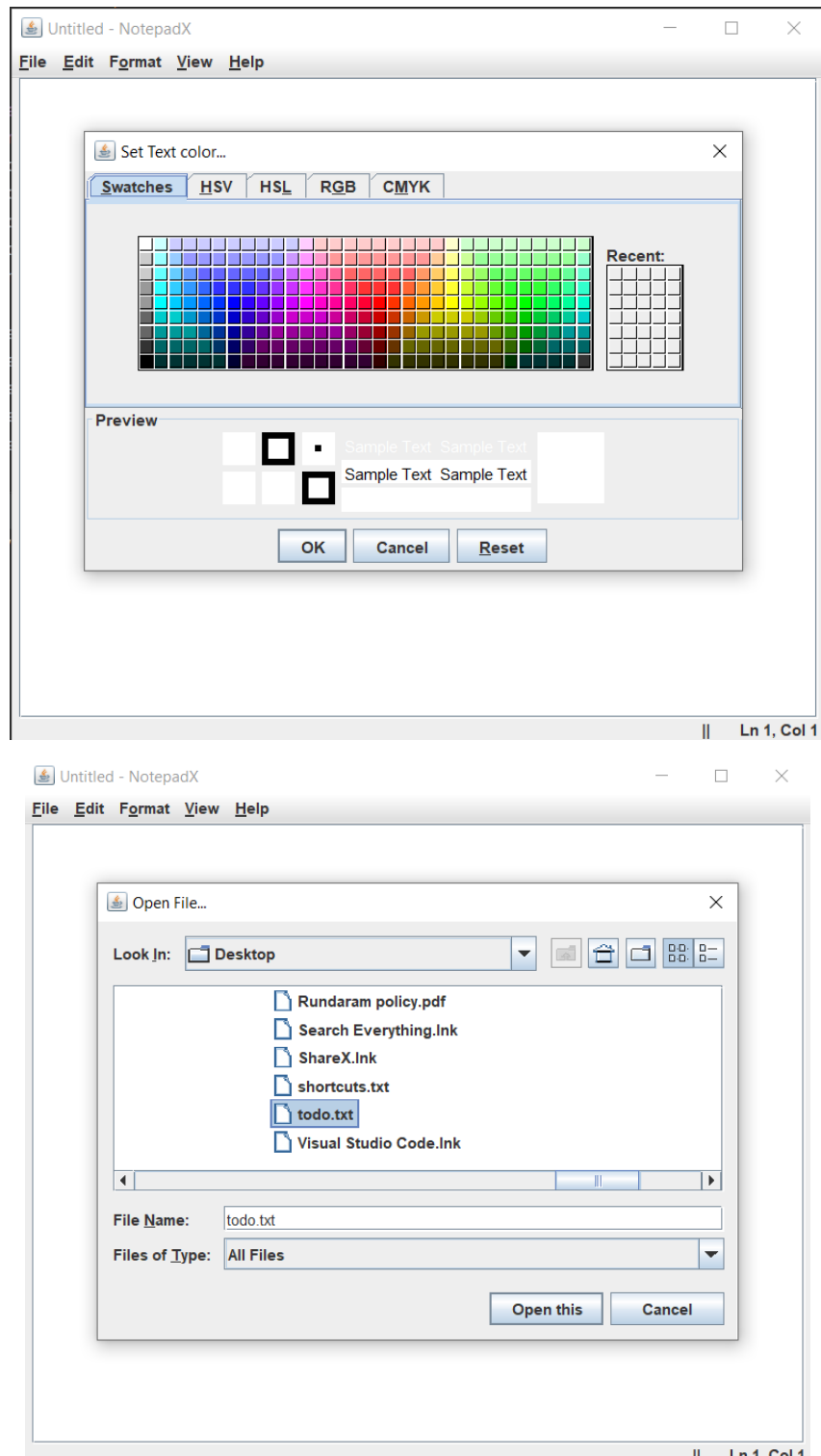
## 6.2 System Testing

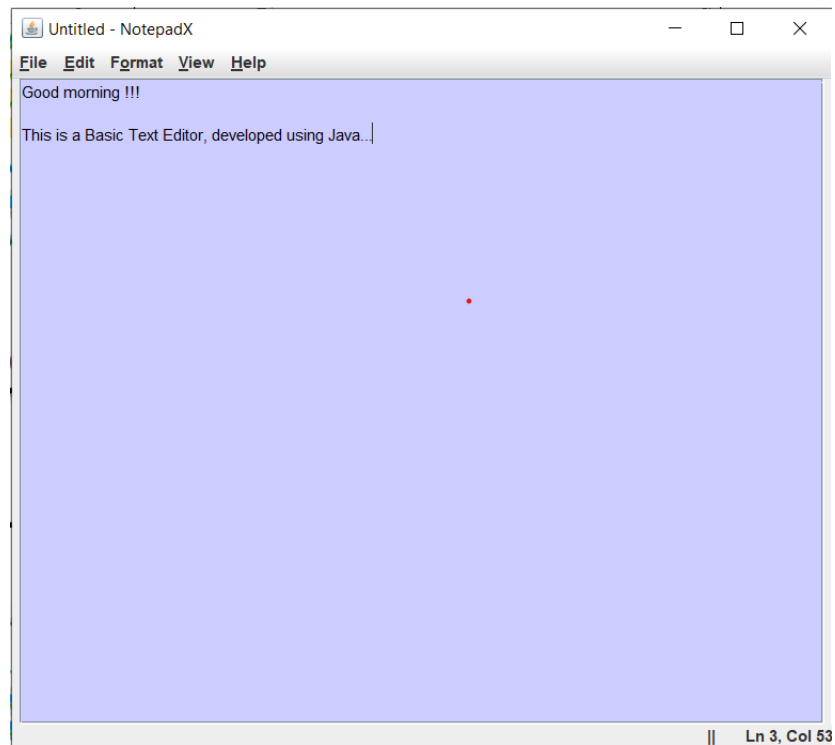
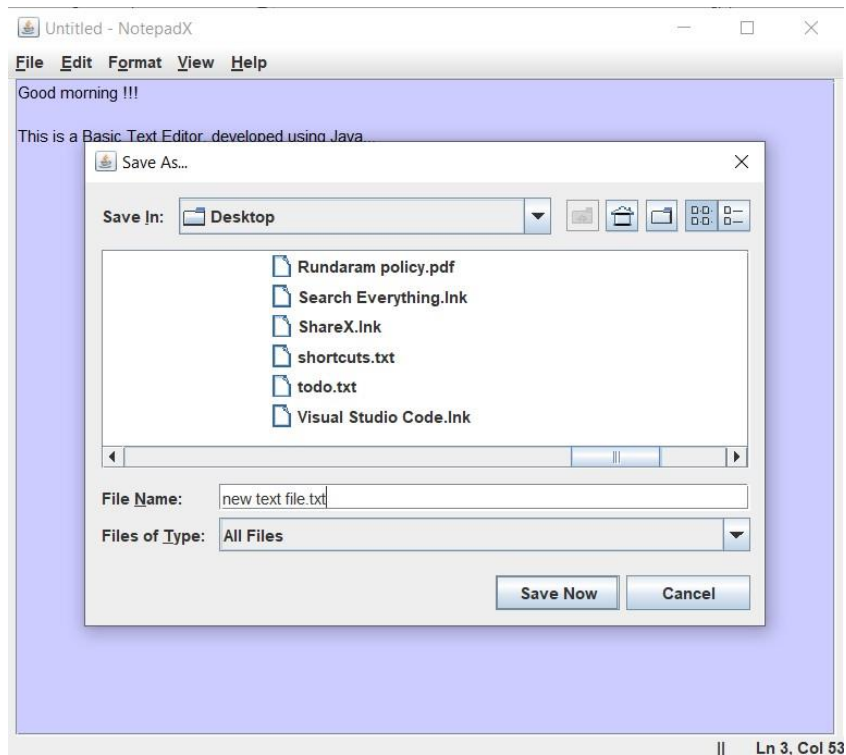
**Table 6.6 System Test**

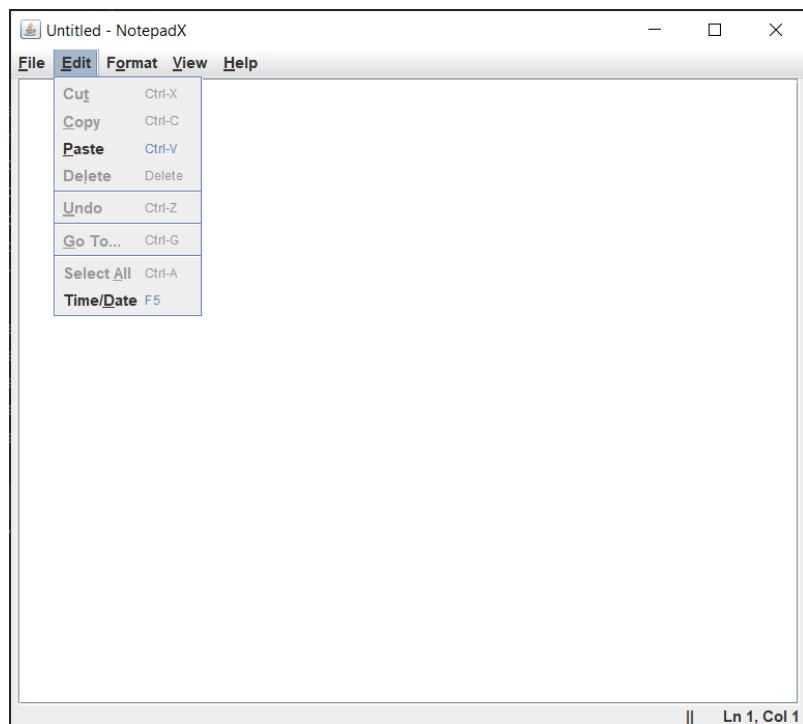
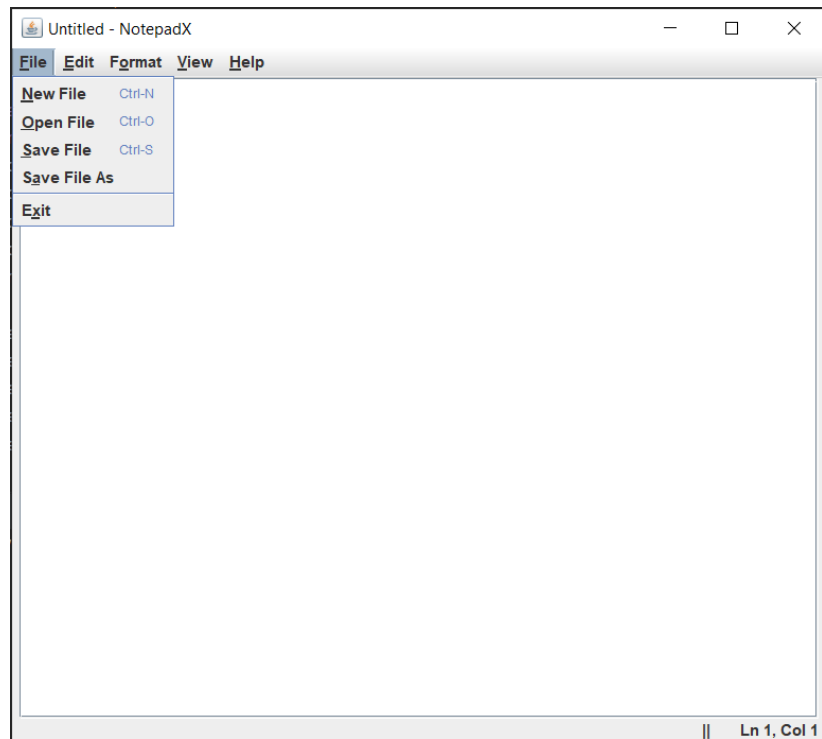
TEST UNIT	TEST CASE	RESULT
Creating a File	Click on New File	Opens up a new fresh window to write text documents
Opening a File	Click on Open File	Opens a previously saved file from storage.
Modifying File Content	On saving a edited and unsaved text file	Modifies Files Content
Saving a File	Click on Save Button	Highlights the corresponding keyword
Changing Text color	Click on Set Text color button in Format Menu	The system will change the color of the text present in the JTextArea.
Changing Background color	Click on Set Pad color button in Format Menu	The system will change the background of the JTextArea.

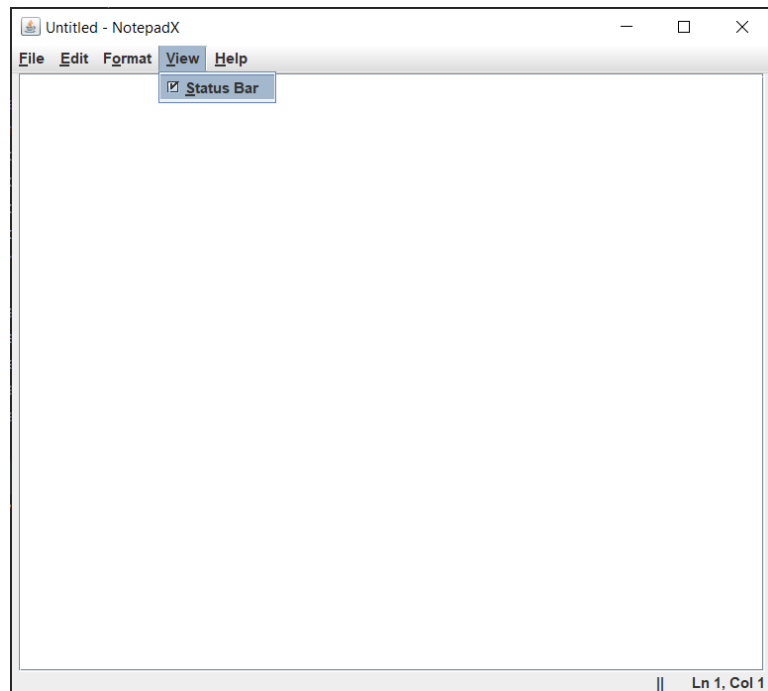
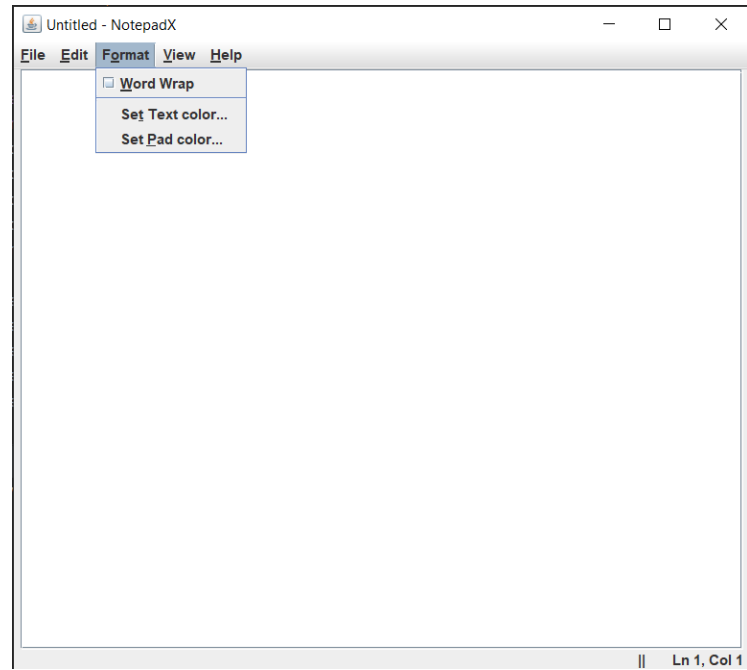
## Chapter 7

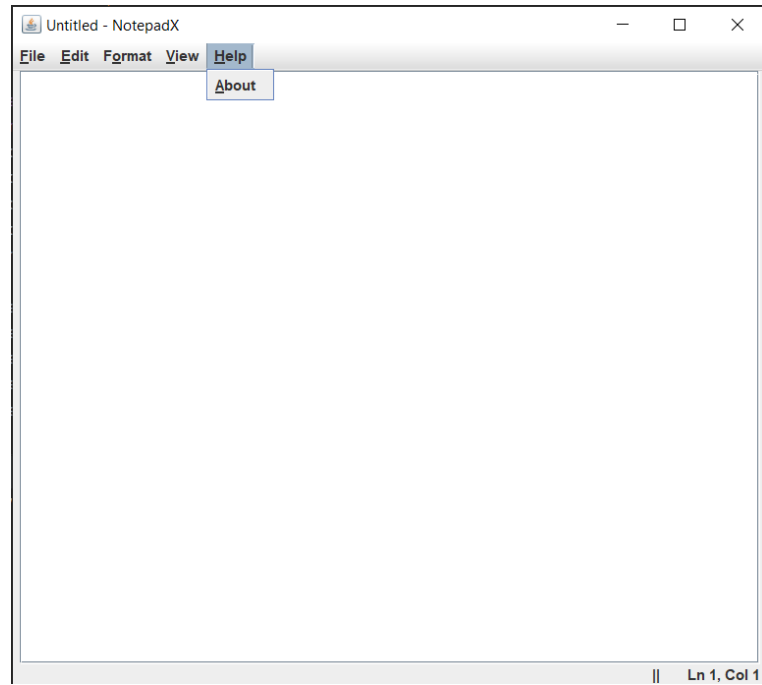
# INTERPRETATION OF RESULTS













## **CONCLUSION**

This project uses the basic concepts of file structure to achieve the objectives of the project. Concepts like Opening, Closing, Reading and Writing of files will be used during implementation.

## **REFERENCES**

- [1]. G. Wallace, R. Biddle and E. Tempero, "Smarter cut-and-paste for programming text editors," Proceedings Second Australasian User Interface Conference. AUIC 2001, 2001, pp. 56-63, doi: 10.1109/AUIC.2001.906277.
- [2]. Yousif, Samar. (2019). DESIGN A SECURE NOTEPAD TO IMPLEMENT A WEBSITE FOR LEARNING JAVA LANGUAGE.
- [3]. <https://docs.oracle.com/en/java/>
- [4]. <https://docs.oracle.com/javase/tutorial/uiswing/>