

PROFESSIONAL TRAINING REPORT

at

**Sathyabama Institute of Science and Technology
(Deemed to be University)**

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology Degree in

Information Technology

By

**K.SOWNDHARYA
(Reg.No.37120073)**



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

**SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

DECEMBER 2020



SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

(Established under Section 3 of UGC Act, 1956)
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai -600119
www.sathyabama.ac.in



DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **K.SOWNDHARYA** (Reg.no.37120073) who carried out the project entitled “**TEXT EDITOR**” under our supervision from JUL 2020 to DEC 2020.

Internal Guide

Dr.R.M.GOMATHI, M.TECH.,Ph.D.

Head of the Department

Dr.R SUBHASHINI M.E,Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I , K.SOWNDHARYA (Reg.No.37120073) hereby declare that the Project Report entitled "**TEXT EDITOR**" done by me under the guidance of **Dr.R.M.GOMATHI, M.TECH.,Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Information Technology .

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing , **Dr.SUBHASHINI M.E., Ph.D.** , Head of the Department of Information and Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.R.M.Gomathi, M.TECH.,Ph.D.** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.



Elite

NPTEL Online Certification

(Funded by the Ministry of HRD, Govt. of India)



This certificate is awarded to

SOWNDHARYA K

for successfully completing the course

Cloud Computing and Distributed Systems

with a consolidated score of **70** %

Online Assignments	24.50/25	Proctored Exam	45/75
--------------------	----------	----------------	-------

Devendra Jaliha

Prof. Devendra Jaliha
Chairman
Centre for Continuing Education, IITM

Total number of candidates certified in this course: 77

Jan-Mar 2020
(8 week course)

Prof. Andrew Thangaraj

Prof. Andrew Thangaraj
NPTEL Coordinator
IIT Madras



Indian Institute of Technology Madras



Roll No: NPTEL20CS48S12390087

To validate and check scores: <https://nptel.ac.in/noc>

ABSTRACT

The project titled Text Editor using python Tkinter. Tkinter is a GUI library from python from which can create multiple GUI apps. Here, using tkinter It develop a notepad like text editor. This notepad will have the menu where can able to create new file, open existing file, save the file, editing, cut and paste, all functionality will there python installed. Text Editor GUI using Python Text editor is an application like notepad to write documents and other files. The text editor is made using python using Tkinter GUI interface. It offers various text editor features such as Functions similar to any other plain text editors, Shows Line Number, Highlights the current line, Shows cursor location at the bottom, open, close,create and edit files. The application is built using python library Tkinter. Forcreating graphic user interface (GUI) on Python we need to use Tkinter library. It also having different packages and approaches for creating GUI like QT for Software Development and Django, Flask for Web app development. Here used Tkinter because it provides a very quick and easy way.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE NO
1	ABSTRACT	
	LIST OF FIGURES	vii
	INTRODUCTION	
	1.1 INTRODUCTION TO THE PROJECT	1
	1.2 AIM AND SCOPE	1
	1.3 FEASABILITY STUDY	1
	1.4 SYSTEMS	3
2	ALGORITHMS AND METHODS	
	2.1 SDLC METHODOLOGY	4
	2.2 ACTIVITY DIAGRAM	6
3	SYSTEM DESIGN AND TESTING PLAN	
	3.1 MODULES	7
	3.2 INPUT DESIGN	8
	3.3 DATA FLOW DIAGRAM	10
	3.4 UML DIAGRAM	11
	3.6 SYSTEM REQUIREMENTS	19
4	RESULTS AND DISCUSSIONS	
	4.1 SUMMARY	22
	4.2 SCREEN SHOTS	23
5	CONCLUSION AND FUTURE ENHANCEMENTS	
	5.1 CONCLUSION	36
	5.2 FUTURE ENHANCEMENT	37
	5.3 REFERENCES	37
6	APPENDIX	
	6.1 SOURCE CODE	38

LIST OF FIGURES

FIGURE NO.	TITLE OF THE FIGURE	PAGE NO.
2.1	Activity Diagram	6
3.1	System Architecture	7
3.3	Data Flow Diagram	9
3.4	Use Case Diagram	10
3.5	Class Diagram	11
3.6	Sequence Diagram	12
3.7	Component Diagram	13

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO THE PROJECT

Tkinter is a Python Package for creating GUI applications. Python has a lot of GUI frameworks, but this is the only framework that's built into the Python standard library. It has several strengths; it's cross-platform, so the same code works on Windows, macOS, and Linux. It is lightweight and relatively painless to use compared to other frameworks. This makes it a compelling choice for building GUI applications in Python, especially for applications where a modern shine is unnecessary, and the top priority is to build something that's functional and cross-platform quickly.

1.2. AIM AND SCOPE

The main aim of the project is "To Create a Text Editor Using Python". Text Editor is the first most and essential need of a software developer those are willing to write code or simply edit the existing one, working becomes easier due to the different functionality provided to the user, they are customizable according to the user needs. As though the ultimate work of a text editor is to edit files then too, choosing the best editor for programming is a trending topic of all-time in the tech industry, depending upon you can predict the power and importance of text editor in developer's life.

1.3. Feasibility Study

1.3.1. Technical Feasibility

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, .at this point in time, not too many detailed design of the system, making it difficult to access issues like performance, costs on (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical analysis.

1.3.2. Operational Feasibility

Proposed project is beneficial only if it can be turned into information systems that will meet the organizations operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed.

Tkinter – This is one of the most Powerful, Widely Used and Platform Independent library available in Python for creating seamless and nice-looking GUI applications, it is very easy to use and understand different widgets provided by Tkinter.

MessageBox – This module provides the template classes and a variety of inbuilt functions for alerting the user, these message boxes can be an info message, an error message or ask yes/no message.

FileDialog – This module provides the classes and inbuilt functions for creating file or directory selection windows.

1.3.3.Economic Feasibility

Economic feasibility attempts to weigh the costs of developing and implementing a new system, against the benefits that would accrue from having the new system in place. This feasibility study gives the top management the economic justification for the new system.

A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be a useful point of reference to compare actual costs as the project progresses. There could be various

1.4 SYSTEMS

1.4.1 EXISTING SYSTEM

The existing system consists of entering the details in the Microsoft Excel Sheets for the storing of the data. When a manager needs information of the employee he searches for the specified file in the file system. He opens the file and takes the information. Report Generation done manually by copying the content of the different files into another file. The Manually generated report was then printed.

DISADVANTAGES OF EXISTING SYSTEM

- Information retrieval is a very big process.
- Lack of organization of the files may lead to information loss due to accidental deletion of files.
- No security because the files are visible to the users.
- Report generation will be a big task.

1.4.2 PROPOSED SYSTEM

The Proposed system is a browser which is completely related to the programming modules so that it has very good performance and it is very easy to deploy in many operating systems.

ADVANTAGES OF PROPOSED SYSTEM

- The performance is increased due to well designed database.
- Security is increased
- Time saving in report generation
- Easy to update the details

CHAPTER 2

ALGORITHM AND METHODS

2.1. SDLC METHODOLOGY

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, "A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

The steps for Spiral Model can be generalized as follows:

- The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
- A preliminary design is created for the new system.
- A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
- A second prototype is evolved by a fourfold procedure:
 1. Evaluating the first prototype in terms of its strengths, weakness, and risks.
 2. Defining the requirements of the second prototype.
 3. Planning an designing the second prototype.
 4. Constructing and testing the second prototype.

- At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involved development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer's judgment, result in a less-than-satisfactory final product.
- The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
- The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
- The final system is constructed, based on the refined prototype.
- The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

The following diagram shows how a spiral model acts like:

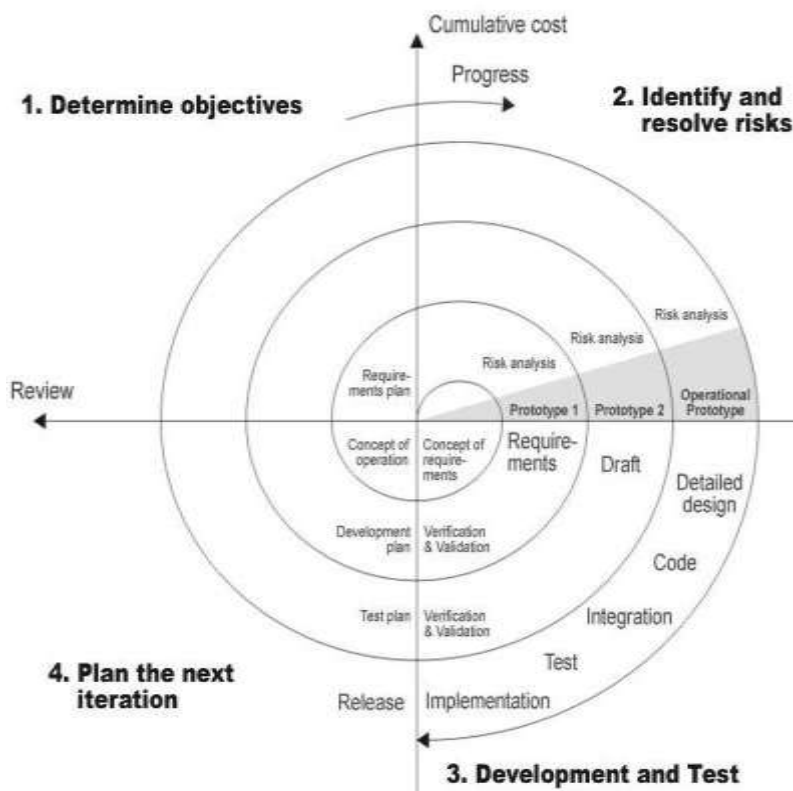


Fig 1.1-Spiral Model

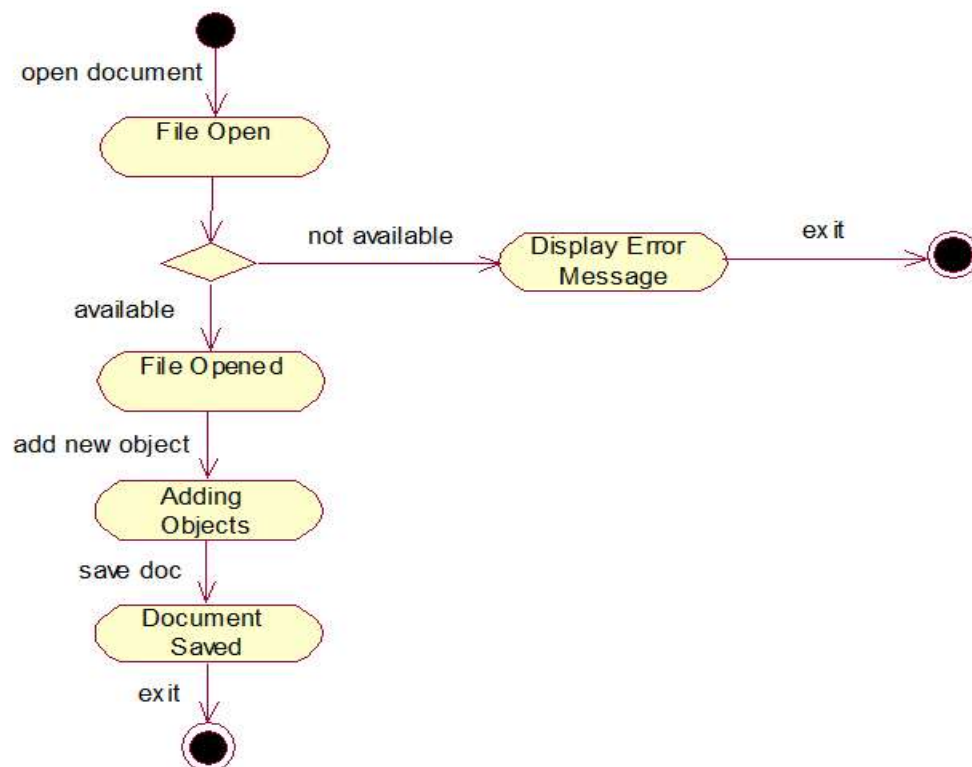
ADVANTAGES

- Estimates(i.e. budget, schedule etc .) become more realistic as work progresses, because important issues discovered earlier.
- It is more able to cope with the changes that are software development generally entails.
- Software engineers can get their hands in and start working on the core of a project earlier.

2.2 ACTIVITY DIAGRAM

ACTIVITY DIAGRAM

Admin



2.1 Activity diagram

CHAPTER 3

SYSTEM DESIGN AND TESTING PLAN

3.1.1 MODULES

ADMIN: This module has the entire access to all other modules, admin creates the project and assigning the projects to the created manager, adding members to the managers, assigning Defects based on the priority. Can update the manager, members and access to the particular project data. Generating reports based on the managers report submission.

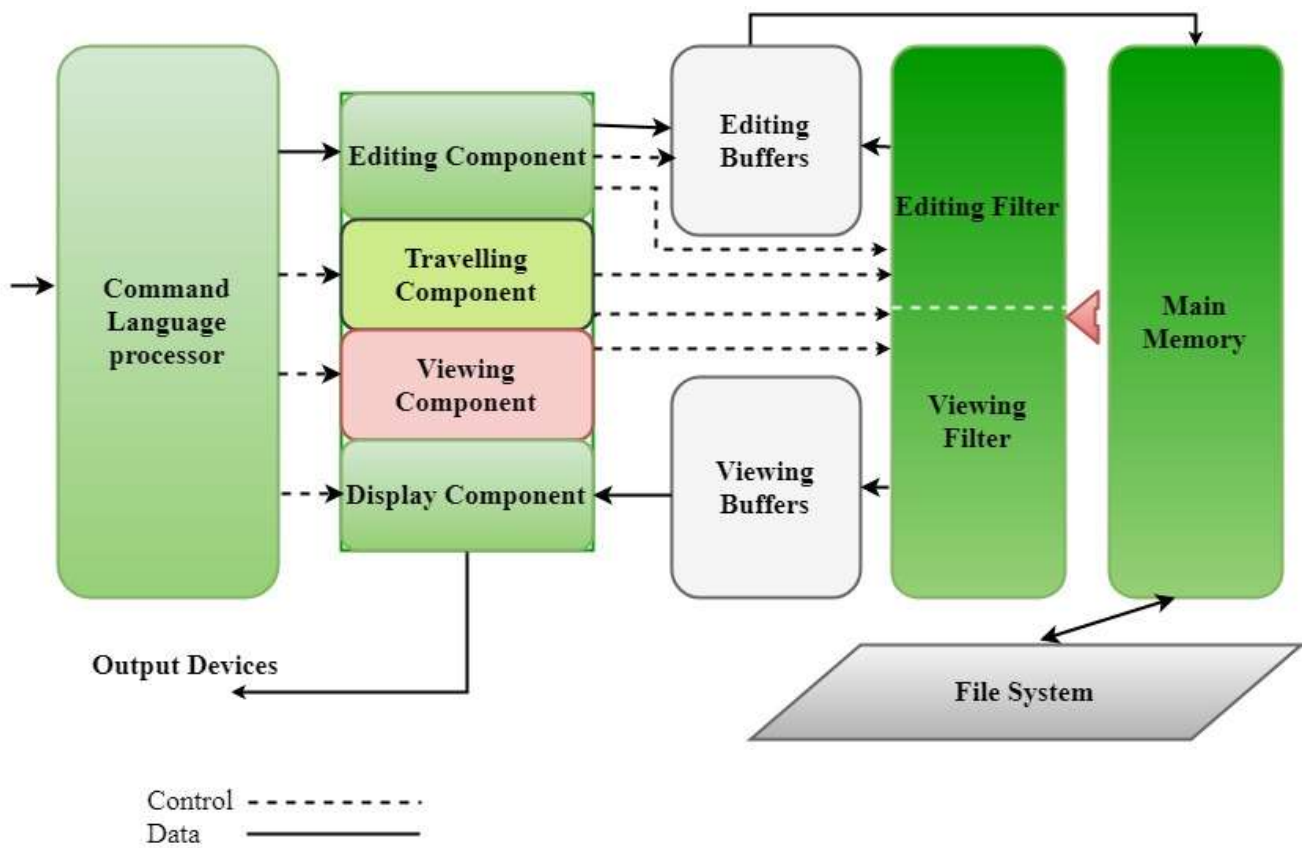
User: Manager has the full access to the particular project assigned by the admin and controls the team members access to the Defects assigned. Has the permission to generate the reports and update the information of team members and adding members to the project.

Developer: Can access the task or Defect assigned by the manager, view assigned projects and resolving the assigned Defect. Developer can view the Defects list assigned by the manager.

Tester: Tester can access to the projects , can view the assigned projects and can add a new Defect to the list and send the bug back to the manager. Tester can login to the system and access the assigned projects list.

Reports: Both Admin and Manager can access this module and generate the reports based on the requirements.

3.1.2 SYSTEM ARCHITECTURE



3.1 Text Editor

3.2 INPUT DESIGN

3.2.1 THE PRESENTATION LAYER

This Also called as the client layer comprises of components that are dedicated to presenting the data to the user. For example: Windows/Web Forms and buttons, edit boxes, Text boxes, labels, grids, etc.

3.2.2 THE BUSINESS RULES LAYER

This layer encapsulates the Business rules or the business logic of the encapsulations. To have a separate layer for business logic is of a great advantage. This is because any changes in Business Rules can be easily handled in this layer. As long as the interface between the layers remains the same, any changes to the functionality/processing logic in this layer can be made without impacting the others. A lot of client-server apps failed to implement successfully as changing the business logic was a painful process.

3.2.3 THE DATA ACCESS LAYER

This layer comprises of components that help in accessing the Database. If used in the right way, this layer provides a level of abstraction for the database structures. Simply put changes made to the database, tables, etc do not affect the rest of the application because of the Data Access layer. The different application layers send the data requests to this layer and receive the response from this layer.

3.2.4 THE DATABASE LAYER

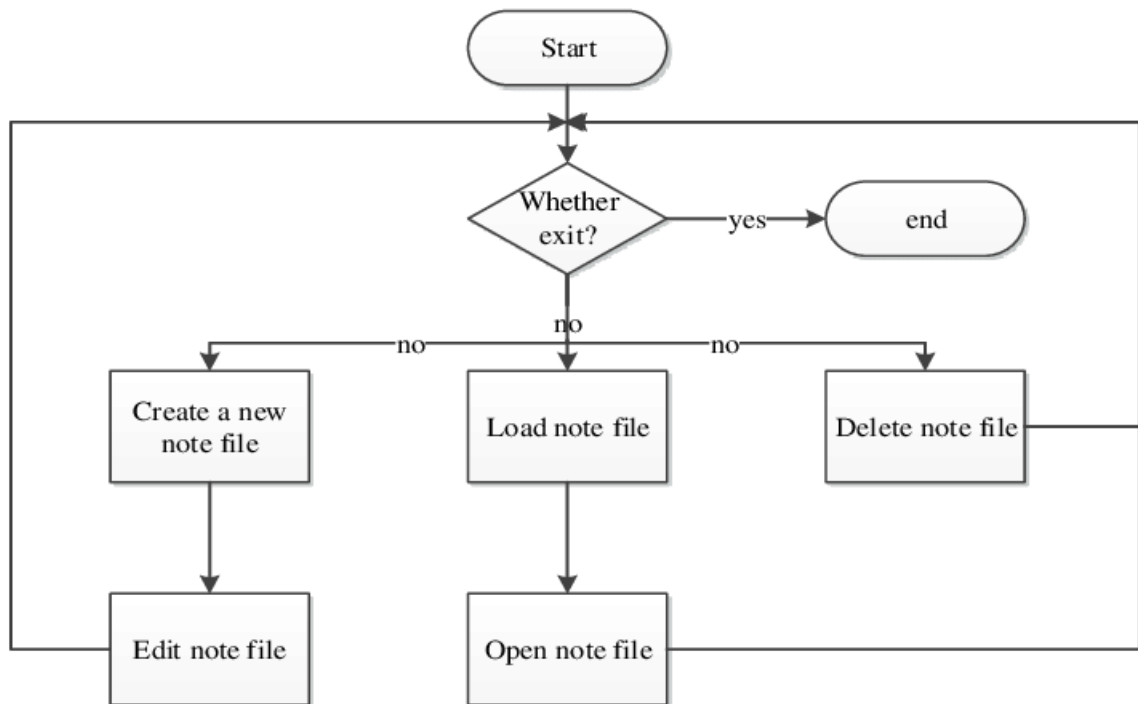
This layer comprises of the Database Components such as DB Files, Tables, Views, etc. The Actual database could be created using SQL Server, Oracle, Flatfiles,etc.

In an n-tier application, the entire application can be implemented in such a way that it is independent of the actual Database. For instance, you could change the Database Location with minimal changes to Data Access Layer. The rest of the Application should remain unaffected.

3.3 DATA FLOW DIAGRAM (DFD)

A data flow diagram is a way of representing a flow of data of a process or a system. The DFD also provides the information about the input and output of each entity and the process itself.

DFD- LEVEL



3.3 data flow diagram

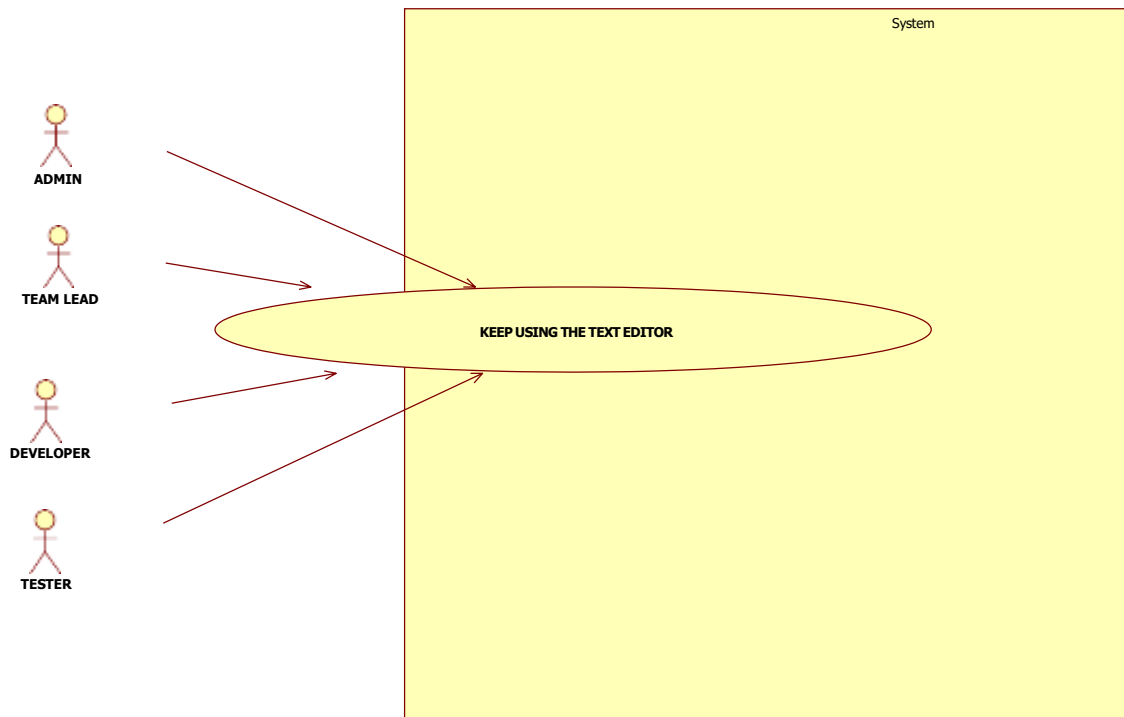
3.4 UML DIAGRAMS

Unified Modelling language includes a set of graphic notations techniques to create a visual models of object oriented software intensive systems

3.4.1 TYPES OF UML DIAGRAMS

1. Use Case Diagram
2. Class diagram
3. Sequential diagram
4. Component diagram

Use case diagram



3.4 Use case diagram

3.5 CLASS DIAGRAM

In software Engineering , a class diagram in the unified modeling language is a type of structure diagram that describes the structure of a system by showing the systems ,classes ,their structures,operations and their relationships among objects.

In the diagram , classes are represented with bases which contain three parts.

1. The top part contains the name of the class.
2. The middle point contains the attributes of the class.
3. The bottom part gives the methods or operations in the class can take or undertake .

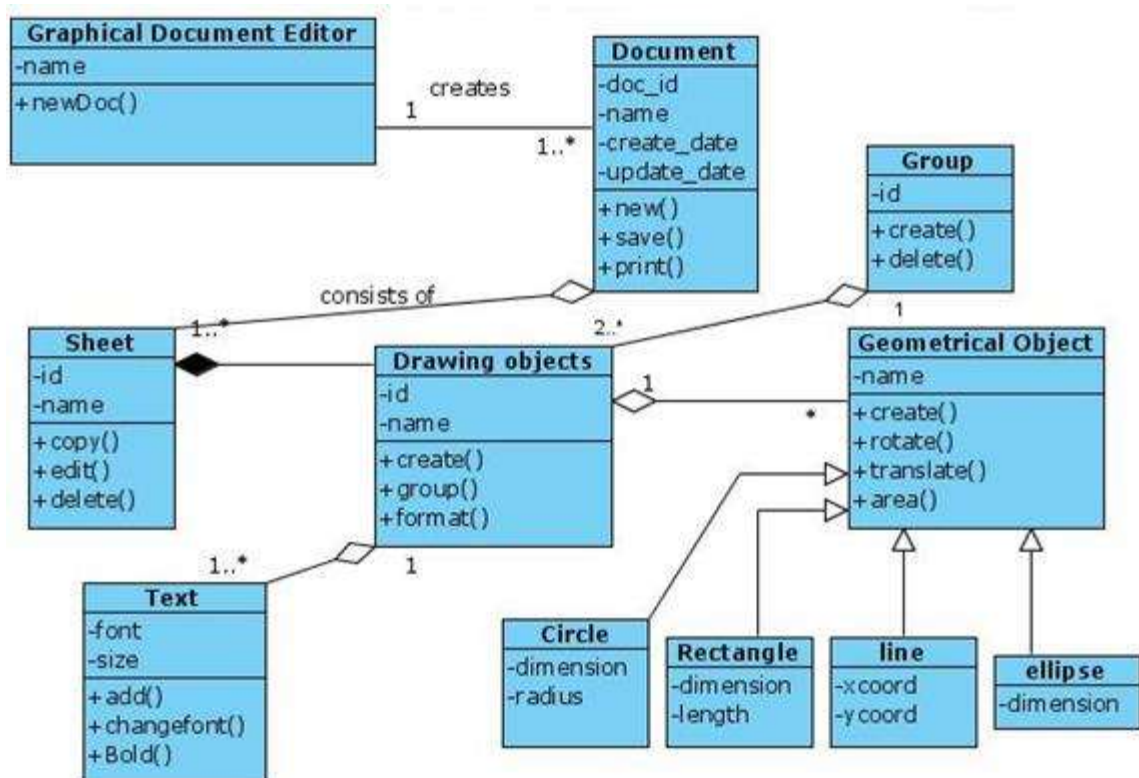
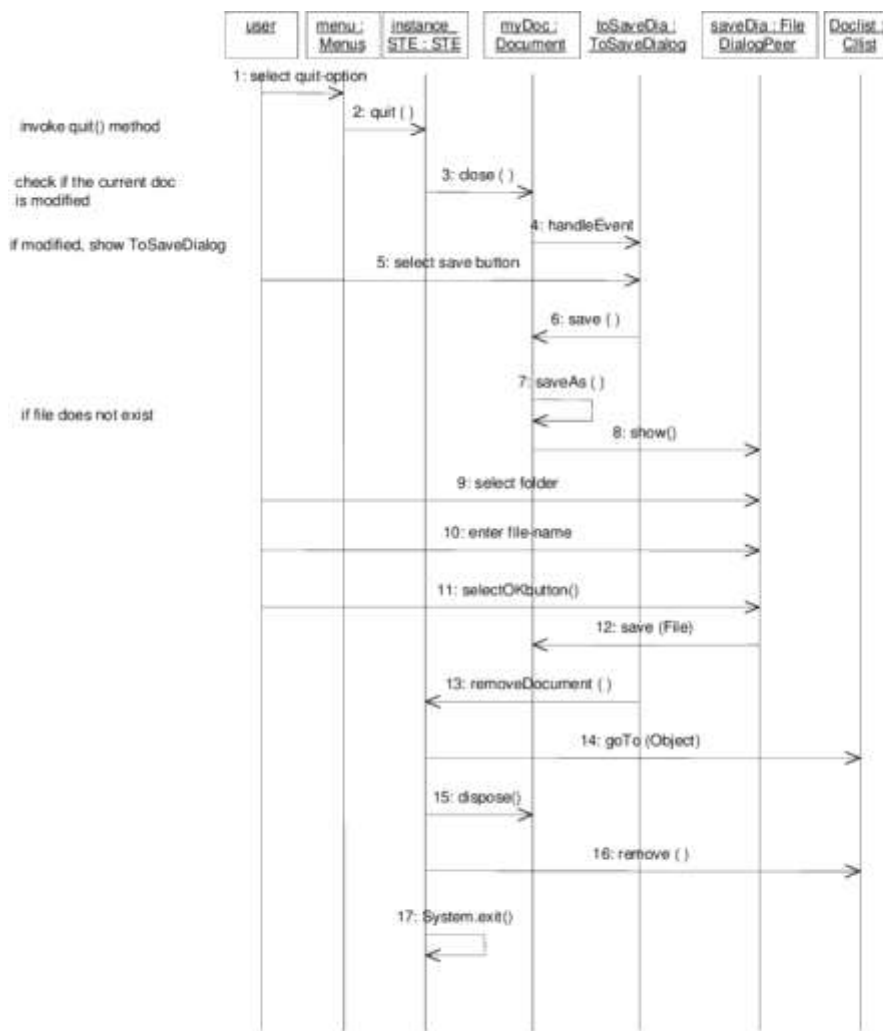


Fig.3.5 class diagram

3.6 SEQUENCE DIAGRAM

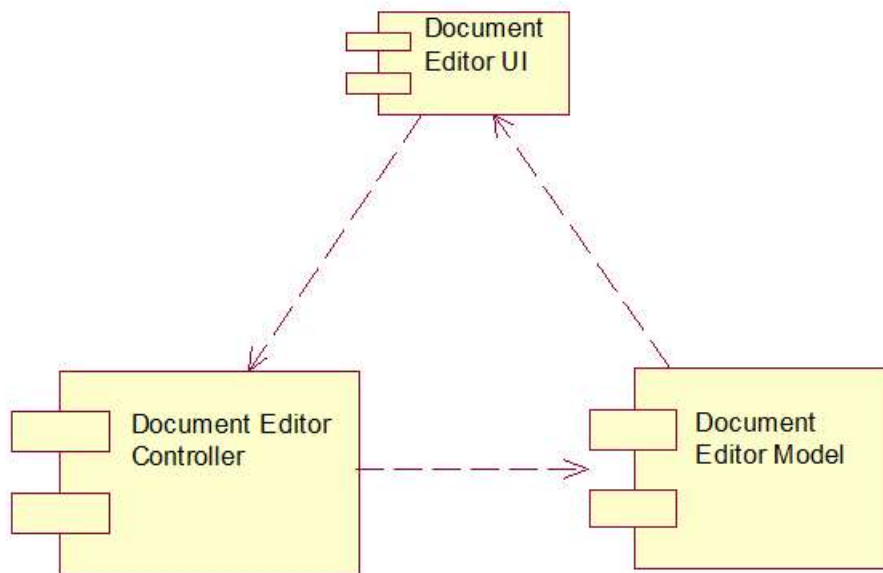
1. The Sequence diagram shows the ordering of the messages .
2. It shows the order of the messages that the admin should perform.
3. The interaction that takes place in a collaboration that either realizes a use case or an operation.
4. High-level interactions between user of the system and the system,between the system and other systems,or between subsystems.



3.6 sequential diagram

3.7 COMPONENT DIAGRAM

1. A component diagram also known as a UML component diagram .
2. It describes the organization and wiring of the physical components in a system.
3. Component diagrams are often drawn to help model implementation details and double check that every aspect of the systems required.



3.7 component diagram

3.8 SYSTEM REQUIREMENTS

3.8.1 Software Requirements

Operating System	:	Windows XP/2003 or Linux/Solaris
User Interface	:	HTML, CSS
Client-side Scripting	:	Python
Programming Language	:	Python
Web Applications	:	JDBC, JNDI, Servlets, JSP
IDE/Workbench	:	Anaconda
Database	:	SQLite
Server Deployment	:	Tomcat6.0

3.8.2 Hardware Requirements

Processor	:	Pentium IV
Hard Disk	:	40GB
RAM	:	256MB

CHAPTER 4

RESULTS AND DISCUSSIONS

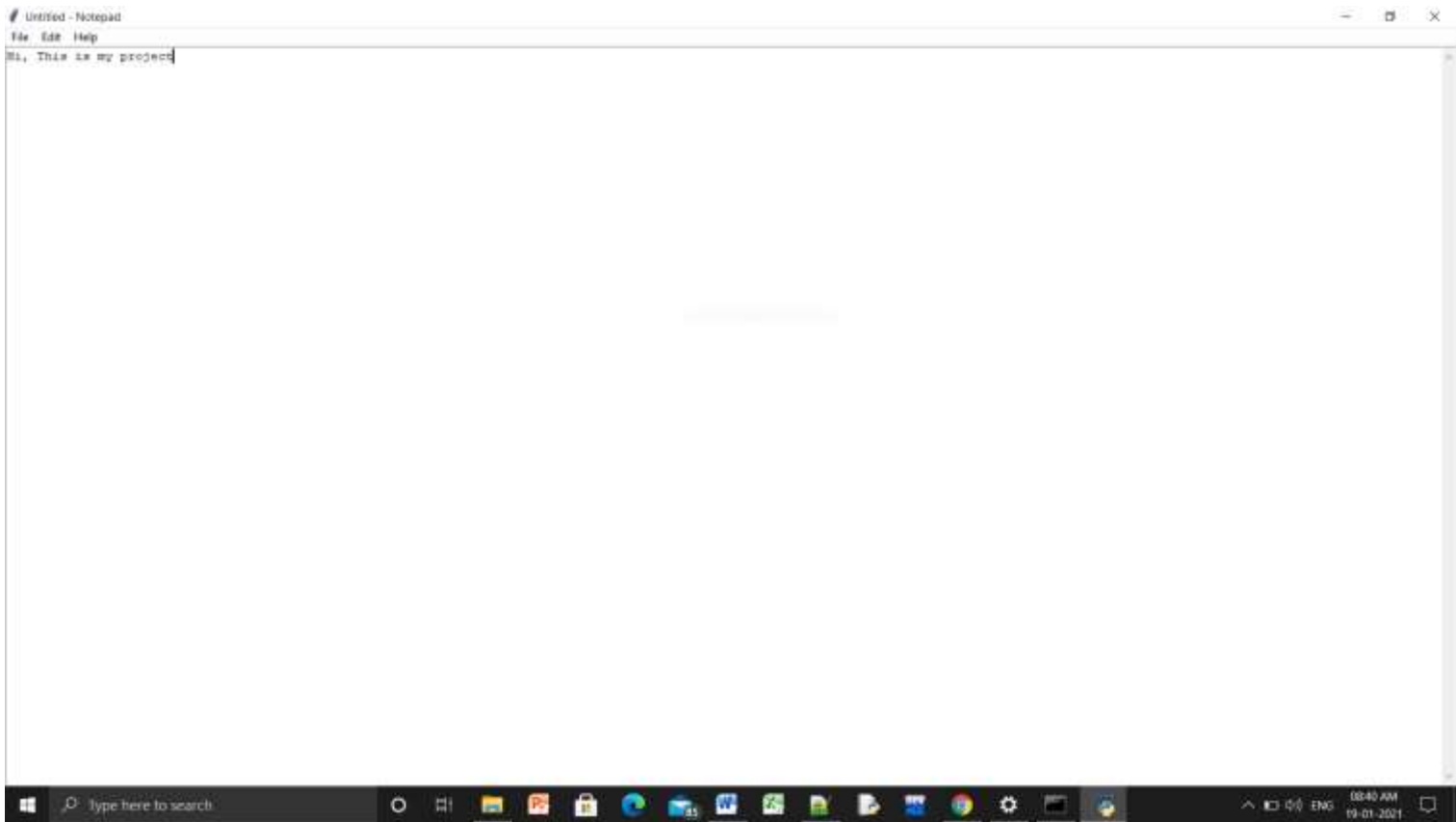
4.1 SUMMARY

Tkinter is a GUI library from python from which we can create multiple GUI apps. Here, using tkinter we will develop a notepad like text editor. This notepad will have the menu where we can create new file, open existing file, save the file, editing, cut and paste, all functionality will there python installed. Text Editor GUI using Python Text editor is an application like notepad to write documents and other files. The text editor is made using python using Tkinter GUI interface.

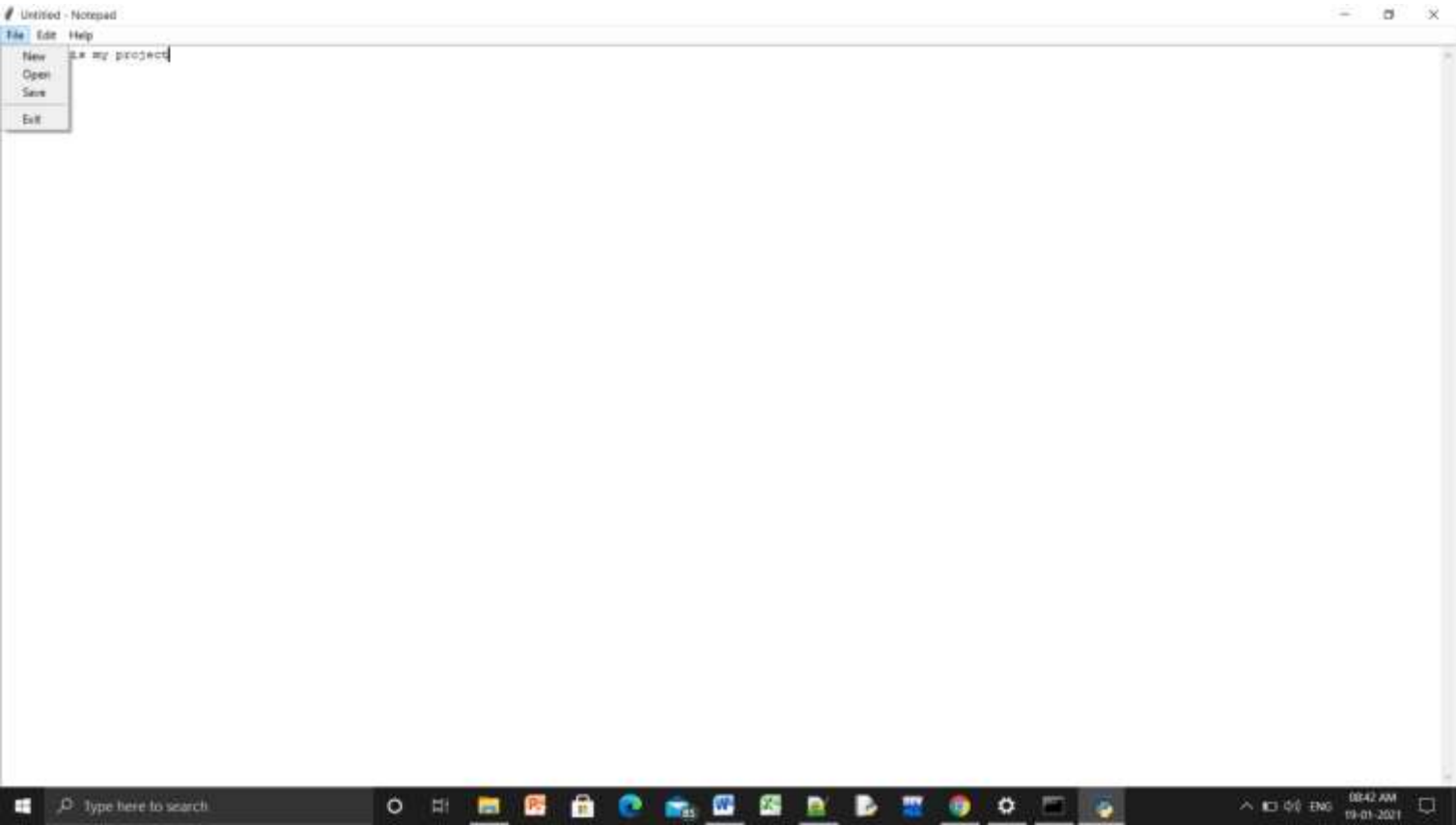
It offers various text editor features such as Functions similar to any other plain text editors, Shows Line Number, Highlights the current line, Shows cursor location at the bottom, open, close, create and edit files. The application is built using python library Tkinter. For creating graphic user interface (GUI) on Python we need to use Tkinter library. We are also having different packages and approaches for creating GUI like QT for Software Development and Django, Flask for Web app development. Here we choose Tkinter because it provides a very quick and easy way to create basic GUI applications.

4.2 SCREEN SHOTS

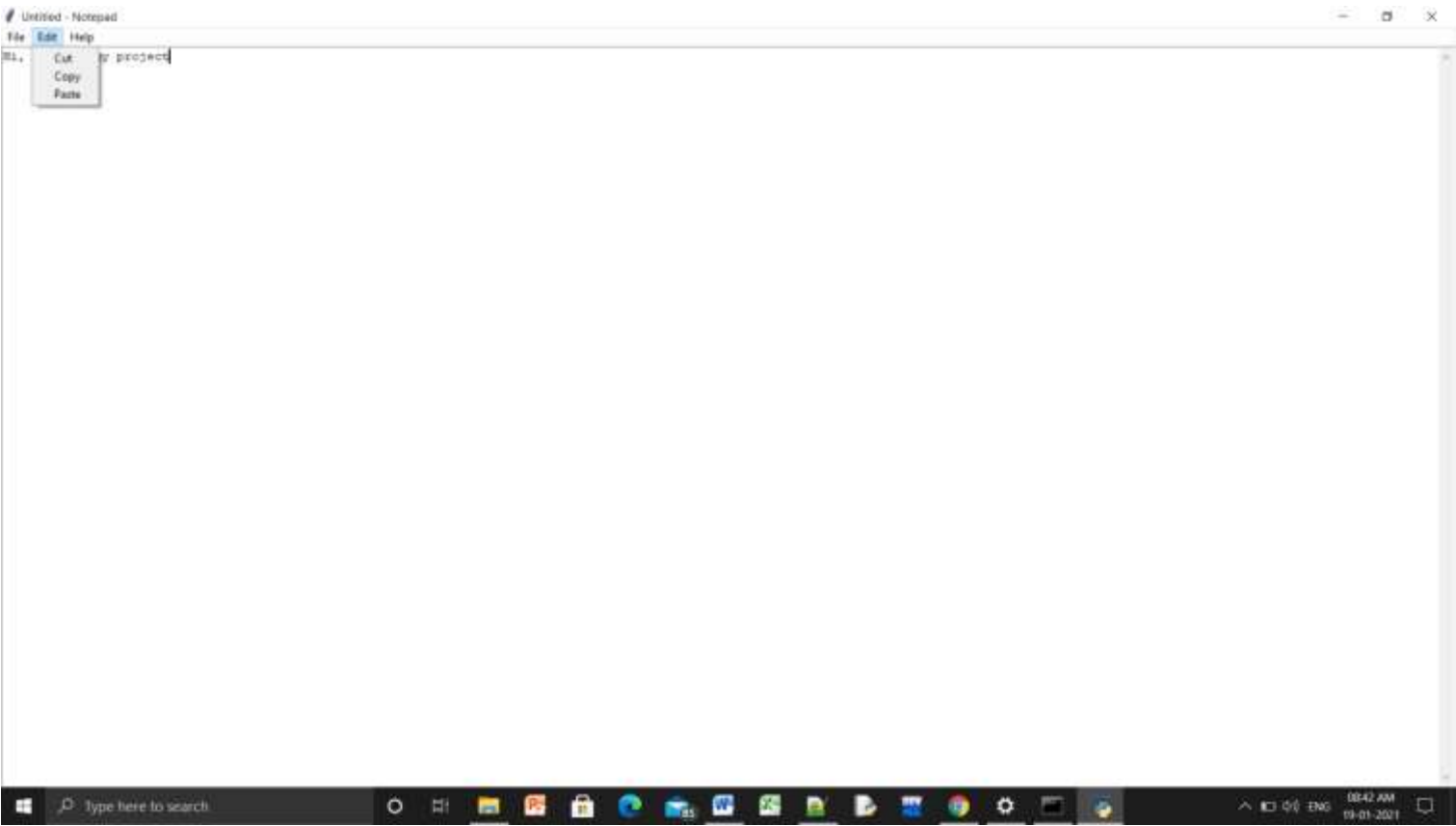
NEW EDITOR WINDOW



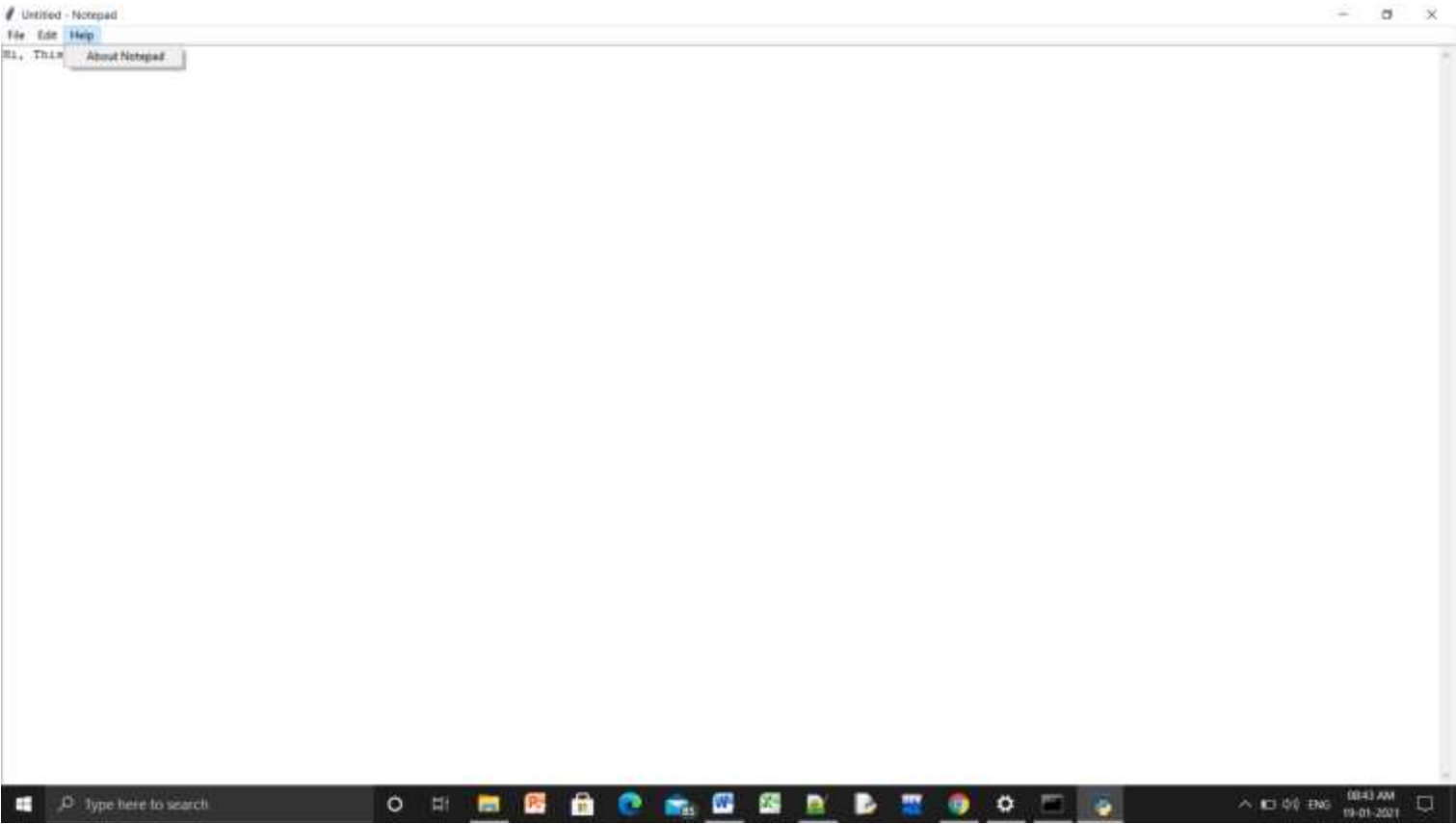
LIST OF MENU IN TEXT EDITOR



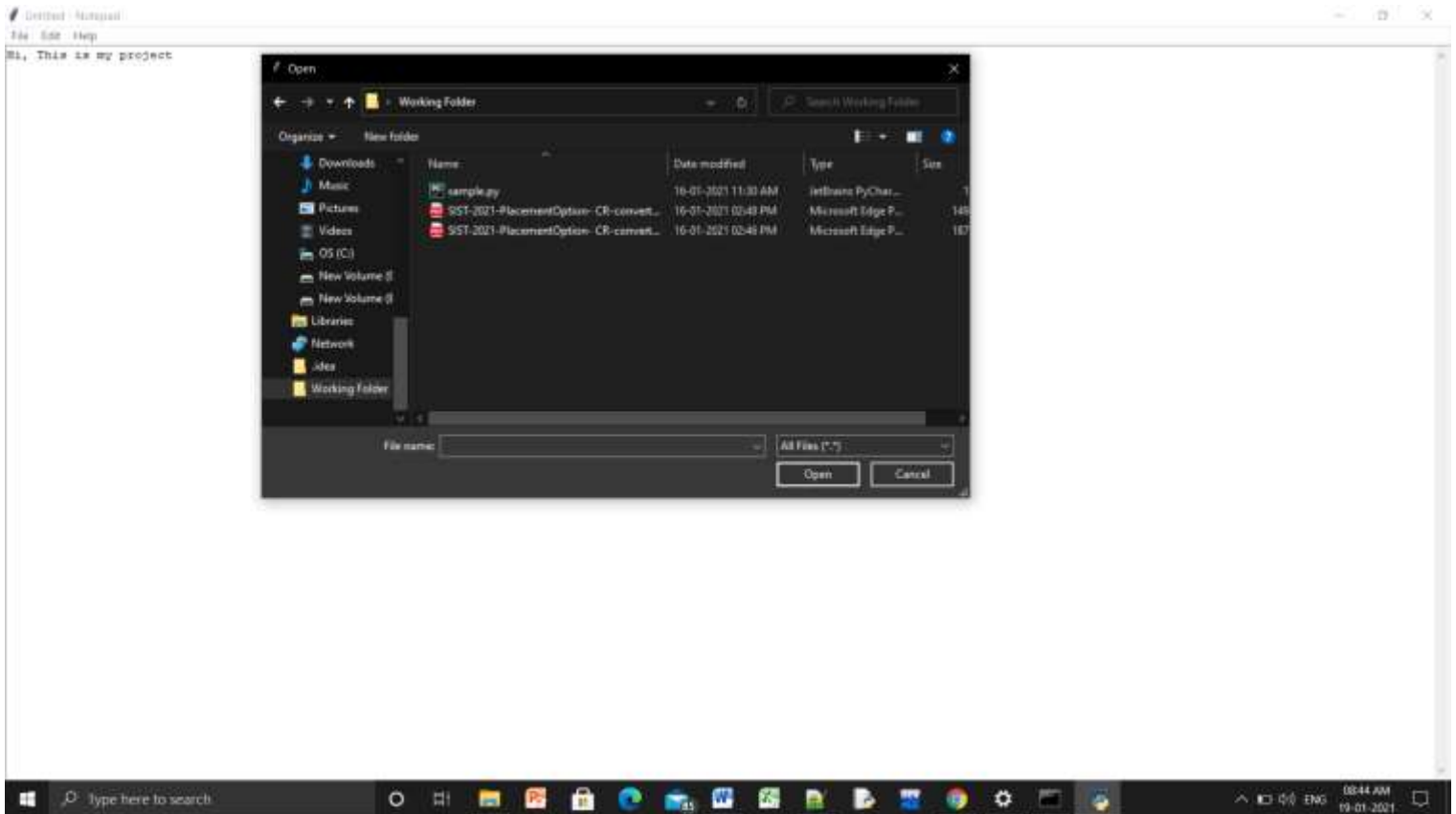
EDIT OPTION



HELP OPTION



SAVE OPTION



CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENT

5.1 CONCLUSION

Tkinter is Python's default GUI library. It is based on the Tk toolkit, originally designed for the Tool Command Language (Tcl). Due to Tk's popularity, it has been ported to a variety of other scripting languages, including Perl (Perl/Tk), Ruby (Ruby/Tk), and Python (Tkinter). The combination of Tk's GUI development portability and flexibility along with the simplicity of a scripting language integrated with the power of systems language gives you the tools to rapidly design and implement a wide variety of commercial-quality GUI applications. Python, along with Tkinter, provides a fast and exciting way to build useful applications that would have taken much longer if you had to program directly in C/C++ with the native windowing system's libraries. Once you have designed the application and the look and feel that goes along with your program, you will use basic building blocks known as widgets to piece together the desired. Once you get Tkinter up on your system, it will take less than 15 minutes to get your first GUI application running.

5.2 FUTURE ENHANCEMENT

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps –

- Import the Tkinter module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

5.3 REFERENCES

1. N. Schuessler and K. Axhausen, "Processing raw data from global positioning systems without additional information," *Transp. Res. Rec., J. Transp. Res. Board*, no. 2105, pp. 28–36, 2009.
2. J. Wolf, "Using GPS data loggers to replace travel diaries in the collection of travel data," Georgia Inst. Technol., Citeseer, Tech. Rep., 2000.
3. J. Ogle, R. Guensler, W. Bachman, M. Koutsak, and J. Wolf, "Accuracy of global positioning system for determining driver performance parameters," *Transp. Res. Rec., J. Transp. Res. Board*, no. 1818, pp. 12–24, 2002.

APPENDIX

6.1 SOURCE CODE

```
import tkinter
import os
from tkinter import *
from tkinter.messagebox import *
from tkinter.filedialog import *

class Notepad:

    __root = Tk()

    # default window width and height
    __thisWidth = 300
    __thisHeight = 300
    __thisTextArea = Text(__root)
    __thisMenuBar = Menu(__root)
    __thisFileMenu = Menu(__thisMenuBar, tearoff=0)
    __thisEditMenu = Menu(__thisMenuBar, tearoff=0)
    __thisHelpMenu = Menu(__thisMenuBar, tearoff=0)

    # To add scrollbar
    __thisScrollBar = Scrollbar(__thisTextArea)
    __file = None
```

```

def __init__(self,**kwargs):

    # Set icon
    try:
        self.__root.wm_iconbitmap("Notepad.ico")
    except:
        pass

    # Set window size (the default is 300x300)

    try:
        self.__thisWidth = kwargs['width']
    except KeyError:
        pass

    try:
        self.__thisHeight = kwargs['height']
    except KeyError:
        pass

    # Set the window text
    self.__root.title("Untitled - Notepad")

    # Center the window
    screenWidth = self.__root.winfo_screenwidth()
    screenHeight = self.__root.winfo_screenheight()

```

```

# For left-align
left = (screenWidth / 2) - (self.__thisWidth / 2)

# For right-align
top = (screenHeight / 2) - (self.__thisHeight / 2)

# For top and bottom
self.__root.geometry('%dx%d+%d+%d' % (self.__thisWidth,
                                         self.__thisHeight,
                                         left, top))

# To make the textarea auto resizable
self.__root.grid_rowconfigure(0, weight=1)
self.__root.grid_columnconfigure(0, weight=1)

# Add controls (widget)
self.__thisTextArea.grid(sticky = N + E + S + W)

# To open new file
self.__thisFileMenu.add_command(label="New",
                                  command=self.__newFile)

# To open a already existing file
self.__thisFileMenu.add_command(label="Open",
                                  command=self.__openFile)

# To save current file

```

```

self.__thisFileMenu.add_command(label="Save",
                                command=self.__saveFile)

# To create a line in the dialog
self.__thisFileMenu.add_separator()

self.__thisFileMenu.add_command(label="Exit",
                                command=self.__quitApplication)

self.__thisMenuBar.add_cascade(label="File",
                                menu=self.__thisFileMenu)

# To give a feature of cut
self.__thisEditMenu.add_command(label="Cut",
                                command=self.__cut)

# to give a feature of copy
self.__thisEditMenu.add_command(label="Copy",
                                command=self.__copy)

# To give a feature of paste
self.__thisEditMenu.add_command(label="Paste",
                                command=self.__paste)

# To give a feature of editing
self.__thisMenuBar.add_cascade(label="Edit",

```

```

        menu=self.__thisEditMenu)

# To create a feature of description of the notepad
self.__thisHelpMenu.add_command(label="About Notepad",
                                command=self.__showAbout)
self.__thisMenuBar.add_cascade(label="Help",
                                menu=self.__thisHelpMenu)

self.__root.config(menu=self.__thisMenuBar)

self.__thisScrollBar.pack(side=RIGHT,fill=Y)

# Scrollbar will adjust automatically according to the content
self.__thisScrollBar.config(command=self.__thisTextArea.yview)
self.__thisTextArea.config(yscrollcommand=self.__thisScrollBar.set)


def __quitApplication(self):
    self.__root.destroy()
    # exit()

def __showAbout(self):
    showinfo("Notepad","Mrinal Verma")

def __openFile(self):

    self.__file = askopenfilename(defaultextension=".txt",

```

```
filetypes=[("All Files","*.*"),
            ("Text Documents","*.txt")]
```

```
if self.__file == "":
```

```
    # no file to open
```

```
    self.__file = None
```

```
else:
```

```
    # Try to open the file
```

```
    # set the window title
```

```
    self.__root.title(os.path.basename(self.__file) + " - Notepad")
```

```
    self.__thisTextArea.delete(1.0,END)
```

```
    file = open(self.__file,"r")
```

```
    self.__thisTextArea.insert(1.0,file.read())
```

```
    file.close()
```

```
def __newFile(self):
```

```
    self.__root.title("Untitled - Notepad")
```

```
    self.__file = None
```

```
    self.__thisTextArea.delete(1.0,END)
```

```
def __saveFile(self):
```

```

if self.__file == None:
    # Save as new file
    self.__file = asksaveasfilename(initialfile='Untitled.txt',
                                    defaultextension=".txt",
                                    filetypes=[("All Files", "*.*"),
                                                ("Text
Documents", "*.txt")])

if self.__file == "":
    self.__file = None
else:

    # Try to save the file
    file = open(self.__file, "w")
    file.write(self.__thisTextArea.get(1.0, END))
    file.close()

    # Change the window title
    self.__root.title(os.path.basename(self.__file) + " - Notepad")

else:
    file = open(self.__file, "w")
    file.write(self.__thisTextArea.get(1.0, END))
    file.close()

def __cut(self):

```

```
self.__thisTextArea.event_generate("<<Cut>>")
```

```
def __copy(self):
```

```
    self.__thisTextArea.event_generate("<<Copy>>")
```

```
def __paste(self):
```

```
    self.__thisTextArea.event_generate("<<Paste>>")
```

```
def run(self):
```

```
    # Run main application
```

```
    self.__root.mainloop()
```

```
# Run main application
```

```
notepad = Notepad(width=600,height=400)
```

