https://www.pes.edu/wp-content/uploads/2019/09/pes_logo.png

**B.TECH. (CSE)**

**V SEMESTER**

**UE20303 –SOFTWARE ENGINEERING**

**PROJECT REPORT**

**ON**

SIMPLE TEXT EDITOR

|  |  |  |
| --- | --- | --- |
| **Sl No** | **Name** | **SRN** |
| **1** | **PRANAV S** | **PES2UG20CS249** |
| **2** | **PRANAV TEJAS K** | **PES2UG20CS250** |
| **3** | **RAJ P KAMBLE** | **PES2UG20CS263** |

**August – Nov 2022**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**BENGALURU – 560100, KARNATAKA, INDIA**

|  |  |  |
| --- | --- | --- |
| **TABLE OF CONTENTS** | | |
| Sl.No | TOPIC | PAGE No |
|  | PROPOSAL OF THE PROJECT | 3 |
|  | SOFTWARE REQUIREMENTS SPECIFICATION | 5 |
|  | PROJECT PLAN |  |
|  | DESIGN DIAGRAMS |  |
|  | TEST CASES |  |
|  | SCREEN SHOTS OF OUTPUT |  |

**SIMPLE TEXT EDITOR**

Project Proposal

Prepared by:

**PRANAV S,PRANAV TEJAS K,RAJ P KAMBLE**

**31/08/2022**

Description:

Introduction:

To develop a text editor in PYTHON that is platform independent that can be easily used by all the users and if there are any changes required then a programmer can easily code and see the changes and it most adaptable.

A text editor is a computer program that lets a user enter, change, store, and usually print text (characters and numbers, each encoded by the computer and its input and output devices, arranged to have meaning to users or to other programs).

Notepad lets you save your notes (as text files) exactly where you want them, and you have no one but your operating system to answer to about how you organize, copy, or back them up.

Plan of work:

1.Select all, Print functions: Pranav S

2.Open, Save, Save as functions: Pranav Tejas K

3.Cut, Copy, Paste, Undo, Redo functions: Raj P Kamble



**PES UNIVERSITY, BANGALORE**

**Department of Computer Science and Engineering**

Software Requirements Specification

**Text Editor**

Using Python(GUI)

**Version 1.0 approved Prepared by PRANAV S**

**PRANAV TEJAS K RAJ P KAMBLE**

**PES UNIVERSITY, BANGALORE**

**Date:16-09-2022**



# Table of Contents

[Table of Contents 2](#_bookmark0)

[Revision History 3](#_bookmark1)

1. Introduction 4
   1. Purpose 4
   2. Intended Audience and Reading Suggestions 4
   3. Product Scope 4
   4. References 4
2. Overall Description 4
   1. Product Perspective 4
   2. Product Functions 5
   3. User Classes and Characteristics 5
   4. Operating Environment 5
   5. Design and Implementation Constraints 7
   6. Assumptions and Dependencies 7
3. External Interface Requirements 8
   1. User Interfaces 8
   2. Software Interfaces 8
   3. Communications Interfaces 8
4. Analysis Models 8
5. System Features 9
   1. System Feature 1 9
6. Other Nonfunctional Requirements 10
   1. Performance Requirements 10
   2. Safety Requirements 11
   3. Security Requirements 11
   4. Software Quality Attributes 11
   5. Business Rules 11
7. Other Requirements 12

[Appendix A: Glossary 12](#_bookmark2)

[Appendix B: Field Layouts 13](#_bookmark3)

[Appendix C: Requirement Traceability matrix 14](#_bookmark4)

Revision History**: Simple Text Editor**

|  |  |  |  |
| --- | --- | --- | --- |
| **Authors** | **Date** | **Description** | **Version** |
| **Pranav S**  **Pranav Tejas**  **Raj P Kamble** | **15-09-2022** | **The final version of the SRS document has been drafted with all the requirements being incorporated into the document** | **1.0** |



## 

## Introduction

### Purpose

To develop a text editor in PYTHON that is platform independent that can be easily used

by all the users and if there are any changes required then a programmer can easily

code and see the changes and it most adaptable.

### Intended Audience

A target audience is the person or group of people a piece of writing is intended to reach.

In other words, it is important for a writer to know who will be reading his or her writing.

This audience is the person or group of people the writer is aiming for or trying to reach

### Product Scope

The scope includes a wide variety of education, entrepreneur, that depend on data . Interactive user interface is the key which helps in the long term of the product. A text editor is a tool that allows a user to create and revise documents in a computer

References MS Word Google Doc Notepad

## Overall Description

### Product Perspective

The product is a full screen text editor with the capacity to edit existing files and create new ones. To make editing simple, we have provided a set of pop down menus in the interface, as well as support for shortcut keys to give commands.



### Product Functions

This editor is a simple editor, very similar to that of a notepad editor that extends the basic features to the end-user like:

1. File open – User can either open the files already existing n the system or open a new blank file.
2. 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.
3. Save As- one can save the file in any desired format like-.txt, .doc, .java etc. The file is stored in any other location specified by the user.
4. Cut-Copy-Paste – This editor also lets the user cut-copy-paste the edited text.

### User Classes and Characteristics

Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.

### Operating Environment

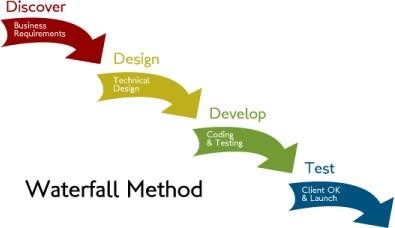
* + Ubuntu 20.04
  + Windows 11
  + Vs code

#### Design and Implementation Constraints Design

Python language is selected for designing this project. Several inbuilt functions and templates present in Python language are used for making the coding easier.

**Waterfall Methodology:**

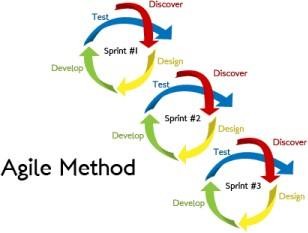
The waterfall methodology is a traditional type of SDLC that functions much like a waterfall when moving from one stage to the next. This greatly reduces flexibility, as when one stage is complete, development moves on to the next stage and rarely goes back. As requirements may change during a project, the waterfall method is unable to deal with this and must either carry on to the end or start over from the planning stage.The waterfall method and how it rigidly mov





**Agile Methodology:**

The agile methodology is used when the requirements may change during the development of a project. Projects using agile development are broken up into sprints, each with their own analysis, design, implementation and testing phases. These sprints begin with a meeting, detailing the work to be done in the sprint with daily meetings to keep the project on target. Essentially, this breaks the project into smaller segments with each segment having its own development phase, then at the end, the segments are combined to form the finished product. Figure 2.2 shows the agile method as a series of sprints with their own development life cycles.



Credits: virtual project environment

For this project, the agile methodology will be utilised. As there is project team, meetings on each sprint is being held, and the method of breaking down the project into smaller, more manageable segments will be used. This will allow the project to be able to adapt to any changes in the strategies of work. As this project consists of a web crawler and a web application, both the agile methodology and waterfall method are better suited to handling multiple deliverables than the single method.

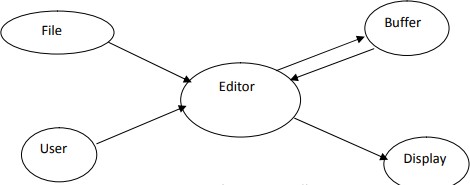
### System Architecture

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.





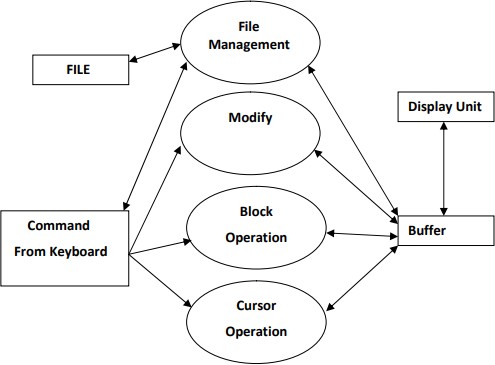
Use Case Diagrams

* 1. Prediction bias might be present.

Implementation Constraints

The system must be reliable enough to run crash and glitch free. Server access issues are also constraints in this software.

Active internet connection.

Windows 7 or above, Ubuntu 16.04 or above is required.

### Assumptions and Dependencies

We assume that the given data in the Official web site of the well-known institution is accurate and correct without any redundancy and false data.



## External Interface Requirements

### User Interfaces

These requirements include user interfaces (interaction logic between software and user), screen layouts, buttons, functions on every screen, hardware interfaces (here a team describes what devices the software is created for), and other relevant particularities.

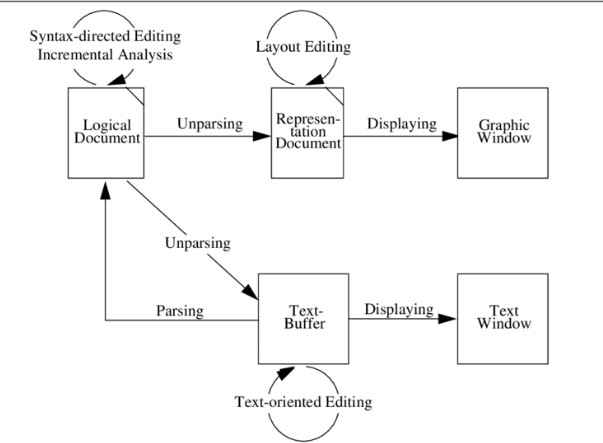
#### Software Interfaces

* + Windows 10/Linux/ubuntu
  + Visual studio code/Notepad++
  + Browsers i.e., chrome, Bing
  + Python 3.8 and above

#### Communications Interfaces

Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.

## Analysis Models





System Features

This editor is a simple editor, very similar to that of a notepad editor that extends the

basic features to the end-user like: File open – User can either open the files already existing n 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. Save As- one can save the file in any desired format like-.txt, .doc, .java etc. The file is stored in any other location specified by the user. Cut-Copy-Paste – This editor also lets the user cut-copy-paste the edited text. Undo – The user is allowed to undo the text edited. This feature allows letter by letter undo. Redo – The user is allowed to redo the text edited. This feature allows letter by letter redo.

#### A Easy to use user interface.

System design refers to formulating a precise description of the desired system in Software Development terms. The information contained in the system specification is similar to that contained in the required documents. The focus is on what functions the Editor must perform, rather than how the Editor performs them. The system specification can be considered as a computer-oriented description of the Requirements Documents. This is represented by a data flow diagram (DFD).

The documentation for the DFD is as follows:

* + - FILE: It is the Floppy Disk/Hard Disk drive where the required source information is stored. This data is to be retrieved into the Primary memory before any changes are done. It is stored in a named file.
* USER COMMAND: This device provides information to modify/append the data. The command is entered through the keyboard/mouse.
* BUFFER: It is the primary memory where the actual modification of information is possible. Data is brought here from the file through commands.
* DISPLAY UNIT: Data can be seen here. It is also called Visual Display Unit. It is possible to see the changes made in this buffer and unit.
* EDITOR: This is the main part of the complete process. It has the actual program to retrieve data from the file and place it in the buffer.



## Other Non-functional Requirements

#### Performance Requirements

* If Some quality characteristics might be incompatible with one another, necessitating trade- offs on the part of the company. Among the several kinds of performance criteria, the followingis one example:
* The capacity to handle a lot of transitions in a short amount of time is knownas high- throughput performance. Higher throughput, however, may compromise reaction time,which measures how soon the user's request is fulfilled by the programmer.
* Another illustration is how the power of the illumination on a visual display can be affected by the predicted battery life of a physical device, possibly affecting usability. Think about how dependencies will affect your performance needs.
* The performance constraints in these individual components must be managed by a software project that depends on OS services or access to an enterprise database.
* Using the necessary designed tools, the document is modified. The view is updated appropriately and saved in this above process, the selection of the part of the document to be viewed and edited involves first traveling through the document to locate the area of interest. Filtering controls the process of what is to be viewed & manipulated.
* Filtering extracts the relevant subset of the target document at the point of interest, such as the next screen of the text or the next statement. In the actual editing phase, the target document is created or altered with a set of operations such as insert, delete, move, copy.
* The editing functions are often specialized to operate on elements meaningful to the type of editor. For example, a manuscript- oriented editor might operate on elements such as single characters, words, lines, sentences, and paragraphs; a program-oriented editor might operate on elements such as identifiers, keywords, and statements.
* In a simple scenario, the, the user might travel to the end of the document. A screenful of text would be filtered, this segment would be formatted, and the view would be displayed on an output device. The user could then, for example, delete the first three words of this view.



#### Safety Requirements

* + Protect the online environment
  + Verify user input
  + Steer clear of external third-party
  + Employ the appropriate verification.

#### Security Requirements

* + Validate user input
  + Avoid third-party scripts and CSS.
  + Use encryption
  + Use the right authentication.

#### Software Quality Attributes

* + Functional suitability.
  + Reliability.
  + Operability.
  + Performance efficiency.
  + Security.
  + Compatibility.
  + Maintainability.
  + Transferability.

#### Business Rules

There Is no as such and until and unless we are using pirated softwares



## Other Requirements

The productivity of editing using full-screen editors (compared to the line based editors) motivated many of the early purchases of video terminals. Some text editors are small and simple, while others offer broad and complex functions. For example, Unix and Unix-like operating systems have the vi editor (or a variant), but many also include the Emacs editor. Microsoft Windows systems come with the simple Notepad, though many people specially programmers—prefer another Windows text editor with more features. Under Apple Macintosh's classic Mac OS there was the native SimpleText, which was replaced under Mac OS X by TextEdit, which merges features of a text editor with those of a word processor such as rulers, margins and multiple font selection. Some editors, such as WordStar, have dual operating modes allowing them to be either a text editor or a word processor.

Text editors for professional users can edit files of arbitrary sizes, such as log files or unusually large texts, such as an entire dictionary placed in a single file. Simpler text editors may just read files into the computer's main memory. On larger files, this may be a slow process, and the entire file might not fit. Some text editors do not let the user start editing until this read-in is complete.

"Programmable editors" can be customized for specific uses. For example, Emacs can be customized by programming in Lisp. One motive for customizing is to make a text editor use the commands of another text editor with which the user is more familiar. A text

editor written or customized for a specific use can sense what the user is editing and assist the user, often by providing simple ways to retrieve related information.

# Appendix A: Glossary

### Factual: Concerned with what is actually the case. Tkinter: Python GUI

MS Word Google Docs Notepad



# Appendix B: Field Layouts

An Excel sheet containing field layouts and properties/attributes and report requirements.

**Sample sheet with information required to register the customer**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Length** | **Data Type** | **Description** | **Is Mandatory** |
| Account Number | 16 | Numeric |  | Y |
| ISFC code | 11 | Alphanumeric |  | Y |
| Card Amount | 20 | Numeric |  | Y |
| Mandate Start Date | 8 | Date | Date of Mandate Registration | N |
| Mandate End Date | 8 | Date | Date of Mandate Expiry | N |
| Status | 25 | Alphanumeric | Status of Registration | Y |
| Customer Name | 60 | String |  | Y |
| Reject Reason Code | 4 | String | Reject Reason code in case mandate is rejected | N |

**Sample Report Requirements: Include the fields to be included in the report**

|  |  |
| --- | --- |
| **Registration Report** | **Transaction Report** |
| Bank Account Number | Transaction Reference Number |
| ISFC Code | Bank Account Number |
| Bank Name | IFSC Code |
| Account Status | Bank Name |
| Account Type | Customer Name |
| Customer Name | Card Number |
| Card Number | Debit Transaction Amount |
| SI Start Date | Transaction Date |
| Status | Status |
| Remarks | Debit Attempt Number |
|  | Remarks |



# 

# Appendix C: Requirement Traceability Matrix

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl. No** | **Requirement ID** | **Brief Description of Requirement** | **Architecture Reference** | **Design Reference** | **Code File Reference** | **Test Case ID** | **System Test Case ID** |
| 1 | Editor | Python and its gui (tkinter) | agile | Notepad | - | 1 | 8 |
|  |  |  |  |  |  |  |  |

**DESIGN DIAGRAM**

Prepared by:

**Pranav S, Pranav Tejas K, Raj P Kamble**

**01/10/2022**

# 

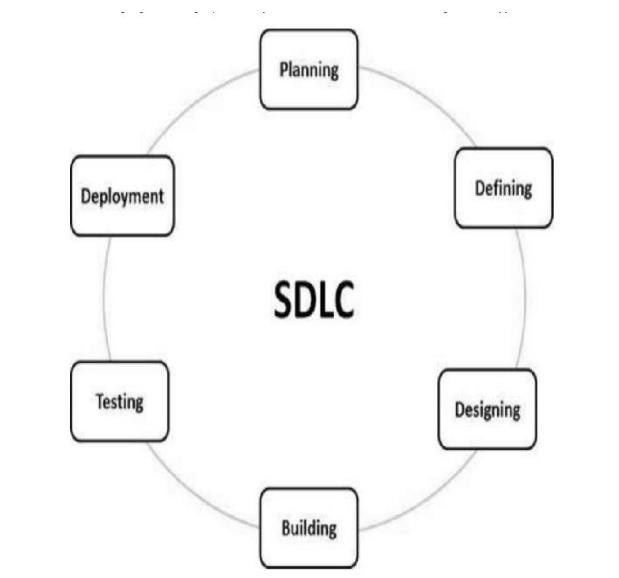
**Project Plan**

**TEXT EDITOR**

1: Identify the lifecycle to be followed for the execution of your project and justify why you have chosen the model.

* 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.
* The systems development life cycle (SDLC), also referred to as the application development life- cycle, is a term used in systems engineering, information systems and software engineering to describe a process for planning, creating, testing, and deploying an information system.
* The systems development lifecycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both
* SDLC enhances development speed and minimizes project risks and costs associated with alternative methods of production
* The agile methodology is used when the requirements may change during the

development of a project



2: Identify the tools which you want to use it throughout the lifecycle like planning tool, design tool, version control, development tool, bug tracking, testing tool.

* For development We used languages like **PYTHON WITH GUI i.e tkinter**

to do this project.

* For version control we used **GitHub.**
* For designing, we used ***JIRA Software.***
* For tracing and testing we used ***JIRA Software.***

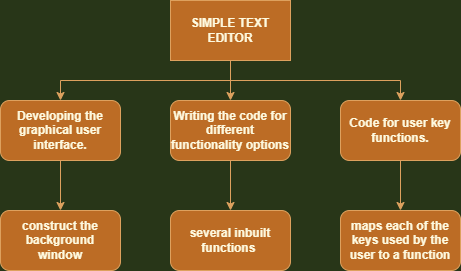
3: Determine all the deliverables and categories them as reuse/build components and justify the same.

***Delivarables :***

* + **A Simple Text Editor**

Create a User Friendly interface with all the features available for a text editor which makes easy handling for the user

Desirable features with its appropriate uses python (gui) 4: Create a WBS for the entire functionalities in detail.



5: Do a rough estimate of effort required to accomplish each task in terms of person months.

Effort, E=ab(KLOC)b

=3.0(0.5)1.12

=1.3802 PM

Time, T=cb(Effort)d

=2.5(1.3802)0.35

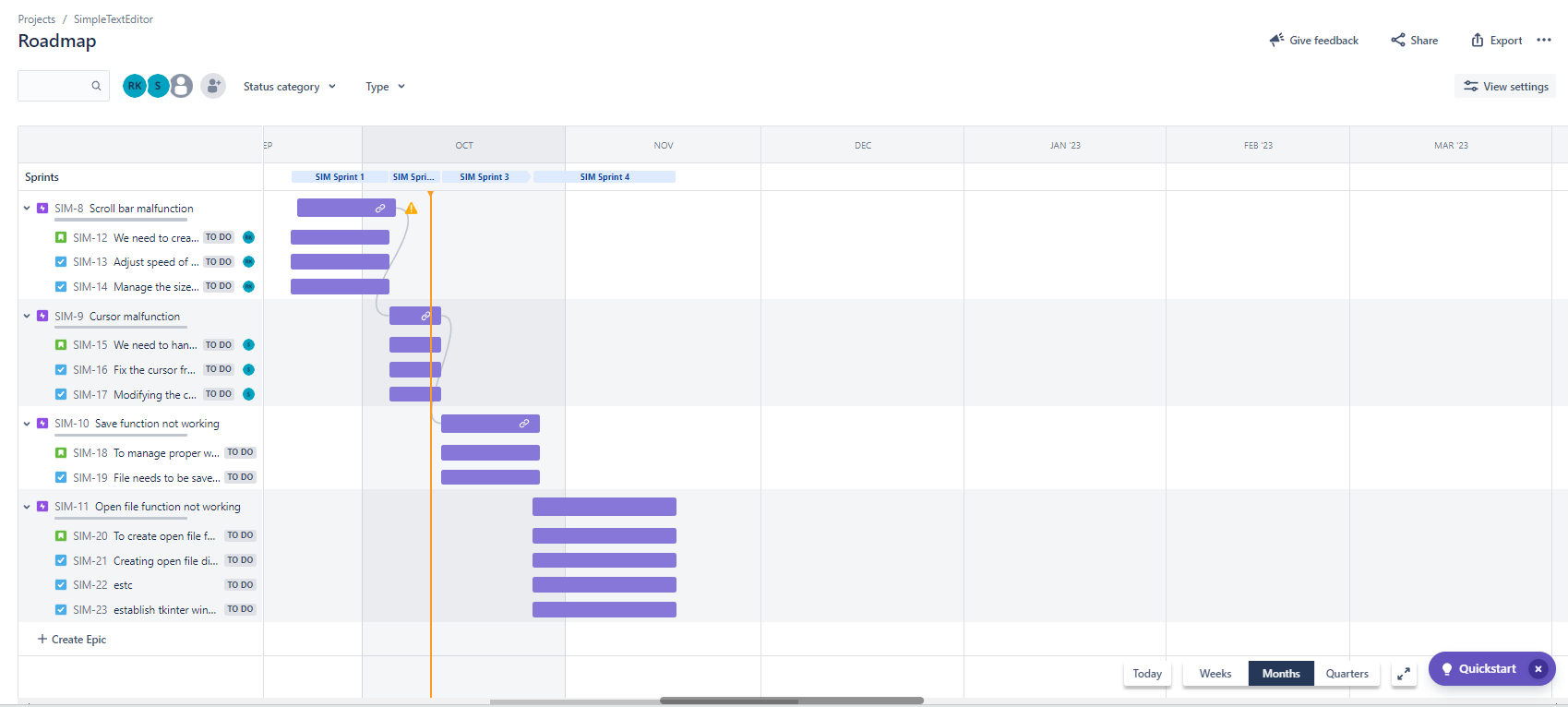
=2.7984 M



PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

6: Create the Gantt Chart for scheduling using any tool.

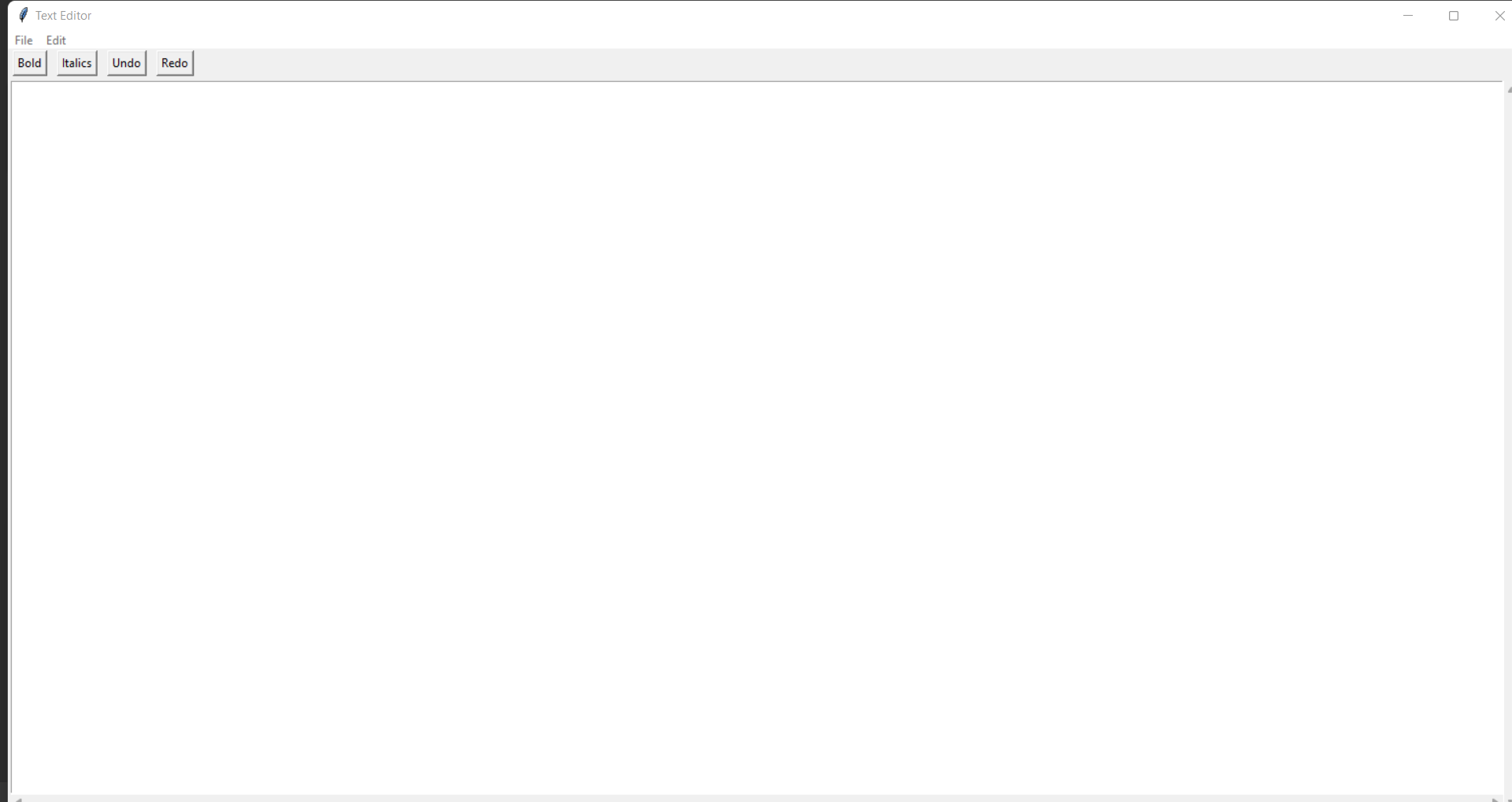
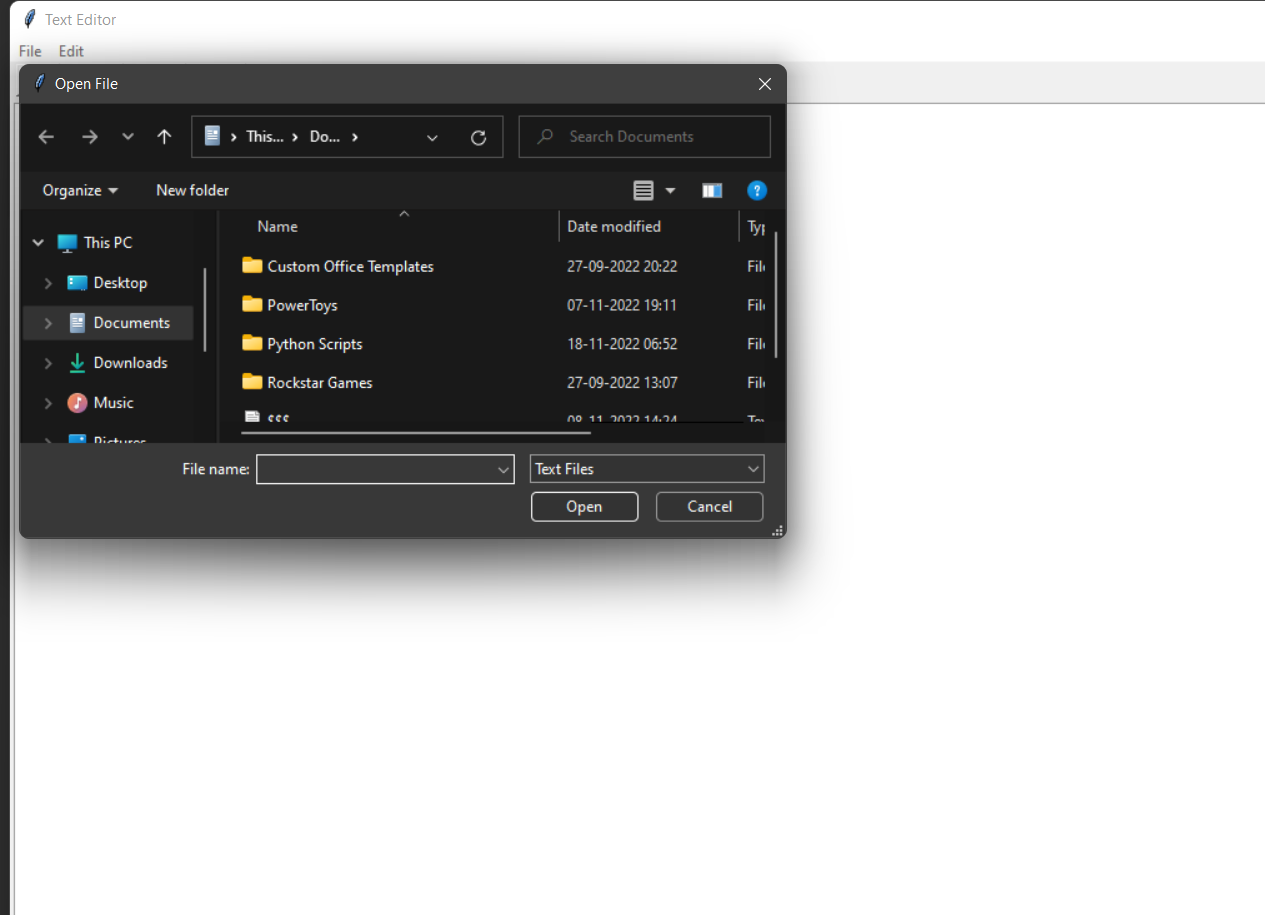


**Test Plan Document**

**Test case:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Name of Module | Test case description | Pre-conditions | Test Steps | Test data | Expected Results | Actual Result | Test Result |
| T1 | SAVE | After the finishing the work the file is saved in a particular | File should be created | Checking if the save button works and opening a dialog box | The file | The file should save in our desired directory successfully | The file saved successfully | Pass |
| T2 | OPEN | Checking if open works which opens a dialog box where we can select it | Only if the particular file does exist | The save button works and opening a dialog box where we can open our desired files | The file having information | Dialog box should open | Dialog box should opened successfully | Pass |
| T3 | UNDO | checking the UNDO | If any task has been done precedingly | any task has been done and erased of and then proceeded with undo | Highlighted text | Any task was done should be erased | It was undo-ed | Pass |
| T4 | REDO | checking the REDO | If any task has been done and wanted to be retrieved | If any task has been undo-ed proceeded by redo button | Highlighted text | Task undo-ed should be back again | Redo was successfully done | Pass |
| T5 | ITALICS and BOLD | checking the ITALICS and BOLD simultaneously | Text should be highlighted | Highlighting the text and proceeded by clicking those respective buttons | Highlighted text | Text should be bold and Italics | The highlighted text were bold and italics respectively | Pass |
| T6 | PRINT | checking the print | If the file is saved and printer has been connected | Saving and clicking the print button | The file having information | It should print the file | The file got printed | Pass |
| T7 | SAVE AS | checking the Save as function opens a dialog box after saving | If the file is saved and wanted to be in different format | Creating the file and clicking the save button | The file having information | It should save as our desired format in our desired directory | It got saved in our desired format successfully | Pass |
| T8 | PASTE | checking the paste is working only if the paste is copied or has been cut | If the text is cut or copied before | Highlight the text and then followed cut/copy and then followed by clicking the paste button or Ctrl-Y | Highlighted text | The text should be pasted successfully | The text has been saved successfully | Pass |

**OUTPUT SCREENSHOTS**

**** ****

