



PES UNIVERSITY RR CAMPUS
SOFTWARE ENGINEERING
UE20CS303

PROJECT TEAM : PR_6

SECTION : 'F'

TOPIC : ADVANCED TEXT EDITOR PYTHON

TEAM MEMBERS :

S/N	NAME	SRN
1	ROHIT ROSHAN	PES1UG20CS355
2	S M SUTHARSAN RAJ	PES1UG20CS362
3	ROHAN C	PES1UG20CS345
4	RAHUL ROSHAN G	PES1UG20CS320



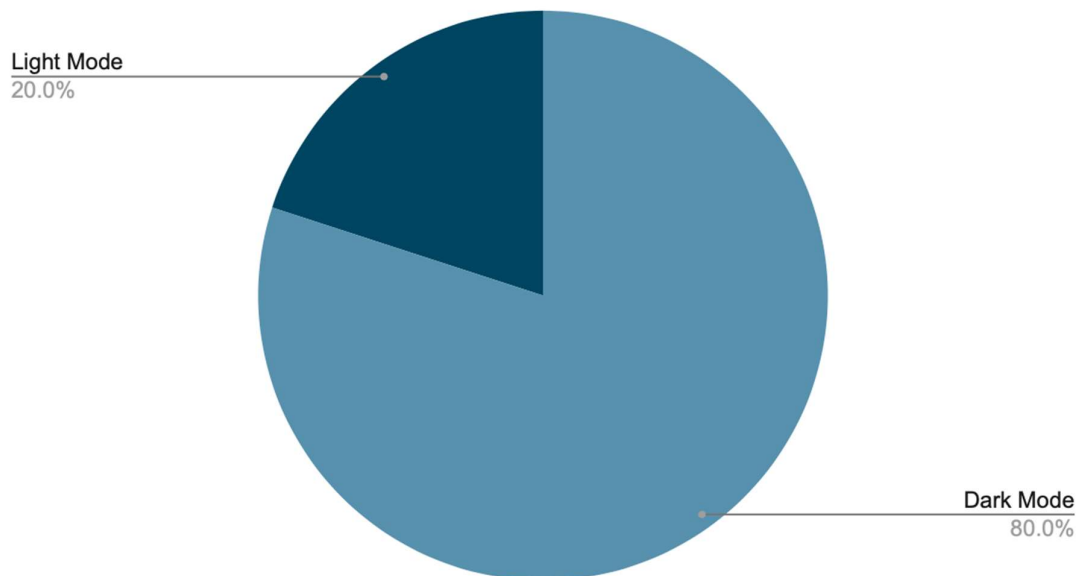
Proposed project description

We have decided to take an existing notepad text editor and improve it into a word processor. This word processor will be able to handle tasks such as 'Find And Replace', 'Chart', 'Diagram', etc.

We will also provide options for developers to code in python. In-built indentation (An important python feature) will also be released in upcoming versions.

User Interface is given a very high priority. By studying customer review and feedback, we have arrived at the following analysis.

User Interface Preference



Hence, it is important for us to incorporate multiple themes for different users to work with.

We have decided to build a complete project and release the product into the market. Most Agile projects are around 3 to 6 months long. If a product needs more development time than this, the product will be released in quarterly cycles -- with one version being released each quarter. So we decided to abide by agile framework because the changes can be handled at any time in the later stages.

The usual users of our software are naive users, python programmers and general programmers and students.

Functional Features:

What the user will be able to perform with your system are listed below:

- Find and Replace.
- Cut, Copy and Paste

- Ability to handle UTF-8 encoded text.
- Text Formatting.
- Undo and redo.
- Line wrap.
- Comment formatting.
- Syntax highlighting.
- Source Code Editor.

Plan of work and product ownership:

At each step of deployment a regression testing phase will be undertaken to make sure that the functionality is working properly.

In this project we are planning to use python. At this stage we are planning and soon we will start with development efforts.

The team would be divided to work on each module. Each of the team members will play separate roles during the development of their modules. They would act as a leader, team members and QA person for different sections of the module.

Once their modules are developed, they will be responsible as a team in performing integration with other modules developed in parallel by other teams.

A feasibility study will be conducted if the deployment of a web-based UI is feasible. If financial, physical and time constraints are satisfied, this feature will also be delivered.

During this process they will be performing integration testing to make sure that all the modules get plugged-in properly.

At each step of deployment a regression testing phase will be undertaken to make sure that the functionality is working properly.

Reference:

1. Common features we are aiming at
<https://haroldpboas.gitlab.io/courses/math696/common-features-of-text-editors.html>

Software Requirements Specifications

Is Crucial To Your Business.

Learn Why!



Software Requirements Specification

Software Requirements Specification

for

TEXT - EDITOR

Version 1.1 approved

Prepared by the team of ‘TEXT EDITOR’

RRRS Pvt. Ltd.

11 - 09 - 2022



Text Editor

Table of Contents

Table of Contents.....	7
Revision History.....	8
1. Introduction.....	9
1.1 Purpose.....	9
1.2 Document Conventions.....	9
1.3 Intended Audience and Reading Suggestions.....	10
1.4 Product Scope.....	10
1.5 References.....	10
2. Overall Description.....	11
2.1 Product Perspective.....	11
2.2 Product Functions.....	11
2.3 User Classes and Characteristics.....	12
2.4 Operating Environment.....	13
2.5 Design and Implementation Constraints.....	14
2.6 User Documentation.....	14
2.7 Assumptions and Dependencies.....	14
3. External Interface Requirements.....	15
3.1 User Interfaces.....	15
3.2 Software Interfaces.....	15
3.3 Communications Interfaces.....	15
4. Analysis Models.....	15
5. System Features.....	18
5.1 System Feature	18
6. Other Nonfunctional Requirements.....	22

6.1 Performance Requirements.....	22
6.2 Safety Requirements.....	23
6.3 Security Requirements.....	23
6.4 Software Quality Attributes.....	24
6.5 Business Rules.....	24
7. Other Requirements.....	24
Appendix A: Glossary.....	25
Appendix B: Field Layouts.....	28
Appendix C: Requirement Traceability matrix.....	28
Team Contribution.....	29

Revision History

Name	Date	Reason For Changes	Version
Synopsis	11-09-2022	To introduce the plan of the text editor.	1.1

1. Introduction

1.1 Purpose



This product “The Text Editor Version no. 1” is a software on which this complete documentation is all about. This document discusses the complete description and working of the end software. All system constraints, functionalities and elements included in the system are explained in detail. By using this documentation, developers can collaborate in order to shape the project to the final specification or they can extend the boundaries of the system for further demand.

This document gives a clear idea of :-

- Usage of the software.
- Prerequisites for the product to work in different computing environments.
- Types of users associated with the product.

This document is divided into 6 different chapters. Each chapter guides you through the description of the product.

1.2 Document Conventions

Formatting (of Document) Conventions:

- The font style for the headings of each section is **Times New Roman** and the font size is 12 for description
- The font style for the headings under each section is **Times New Roman in Bold** and the font size used is 17.
- Coloured fonts are used to describe highlighting components in the documentation.
- Italics or underline has been used to indicate comments and notes.
- Bullet points used to represent steps or sequence.

1.3 Intended Audience and Reading Suggestions

1.3.1 Intended Audience:

This document is primarily intended for the:

- Naïve or common users .
- Coding developers to code for their projects.
- Users who want to work with analytics and represent a graphical output.
- Software developers who can further improve the product.

1.3.2 Document Overview:

- The first chapter, that is the Introduction section of the document is intended to introduce the reader to the product, i.e. The text editor.
- The second chapter, Overall Description section of the SRS document provides an overview of the overall functionality of the product. It describes the informal requirements.
- The third chapter, Specific Requirements section, is for the developers to further understand the technical specification of the product.
- The fourth chapter deals with the external interface requirements.
- The fifth & sixth chapter deals with other non – functional requirements and miscellaneous requirements.

1.4 Project Scope

This software helps people mainly in :-

- Writing texts and including flow-charts and clip arts.
- Writing codes and highlighting syntax.
- Providing different views and settings depending on the user.
- Option to save the files in the format they like.

This tool helps users in various ways, like preparing a list, or helps for documentaries. Apart from that it also helps users to perform many utilities like searching, find and replace, text-highlighting, etc...

1.5 References

<https://youtu.be/CCj1mG2jWKM>

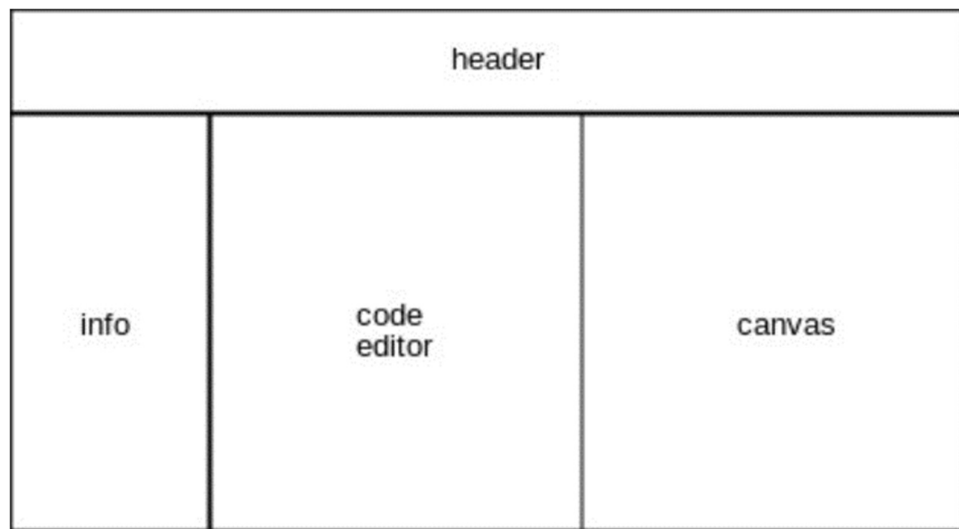
<https://youtu.be/mFdGV8C9o1k>

<https://youtu.be/z-3sklYjVbA>

2. Overall Description

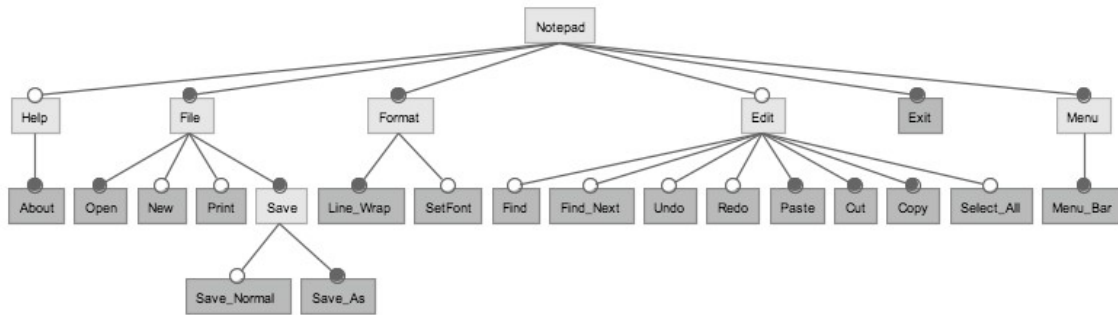
2.1 Product Perspective

This text editor software will help people to type in a GUI view and gives easy access to use various functionalities.



The software aims to be primarily divided as shown in the above figure. This text-editor is very simple for users who are not even familiar with the usage of computer systems. Since this is menu driven, users will have a lot of options confronting them. The functionalities by this software are very professional such as highlighting text, spell-check, inserting flowcharts and cliparts and many more in addition to naïve texting.

2.2 Product Features



This product itself speak for its many of the functionalities such as :-

- · File operations to save, open, etc...
- · Cursor motion and highlighting.
- · Cut and paste
- · Search and replace
- · Customization and coding framework.
- · Clip arts and flow charts

2.3 User Case, Classes and Characteristics

2.3.1 User classes



The system will support these user privileges :-

- Developer
- Naïve user
- Illustrator
- Future software developer to add more functionalities to this editor
- Product owner to decide the future capabilities and investments to this product.
- Testers, for rigidity.

2.3.2 User cases and characteristics

In order to present what these use cases are actually for, their overall roles in the project, the actors related to them and the relations between them are briefly explained in the table below.

Use-case Name	Description
Develop	Develop a custom DGP algorithm in the environment. IDE offers a robust text editor for this purpose.
Step-by-step Debug	Detect the logical errors the custom algorithm may have by classic stepby-step debugging technique
Visual Debug	Detect the logical errors the custom algorithm may have by visual debugging technique
Compare algorithm	Compare the behaviour of the custom algorithm with a known, already implemented one
Use DGP API	Develop a custom DGP algorithm in the environment by an offered, easy-to-use API
Customise IDE	Change predefined settings for the IDE (through preferences) for the suitable working environment

2.4 Operating Environment

The software will be designed to work on any version of Windows, Linux (kernel 2.7 and above) and Mac platform.

2.5 Design and Implementation Constraints



This software enables different views for different varieties of users. Like, for example, an user who simply types can use the regular views , for developers, a coding view, etc.... Therefore, the design is implemented such that it is flexible.

Some of the constraints are :-

- Editor should be a use-once environment. That is, only user preferences are held in the file system, therefore only the changes in preferences are remembered between two successive runs.
- Editor should not let access to calls for all legacy libraries through its lifetime.
- Editor should have one scene at a time.
- Editor should always be initialised to an empty scene.
- Editor should not depend on the operating system.

2.6 User Documentation

Tutorials will be provided in the upcoming versions, which includes various product descriptions.

2.7 Assumptions and Dependencies

Assumptions

The user is familiar with normal typing with the text editor and knows the basic functionalities and settings of any text editor.

Dependencies : As mentioned in the operating environment and demands decent hardware specifications.

3. External Interface Requirements

3.1 User Interfaces

The text enable provides the following UI features:

- a) Find and Replace.
- b) Cut, Copy and Paste.
- c) Ability to handle UTF-8 encoded text.
- d) Text Formatting.
- e) Undo and redo.
- f) line wrap.
- g) comment formatting.
- h) syntax highlighting.
- i) Source Code Editor.

3.2 Hardware Interfaces

The editor can be used in all Personal computers (PC) and Laptop Notebooks or any other system that supports Heroku to run the App.

3.3 Software Interfaces

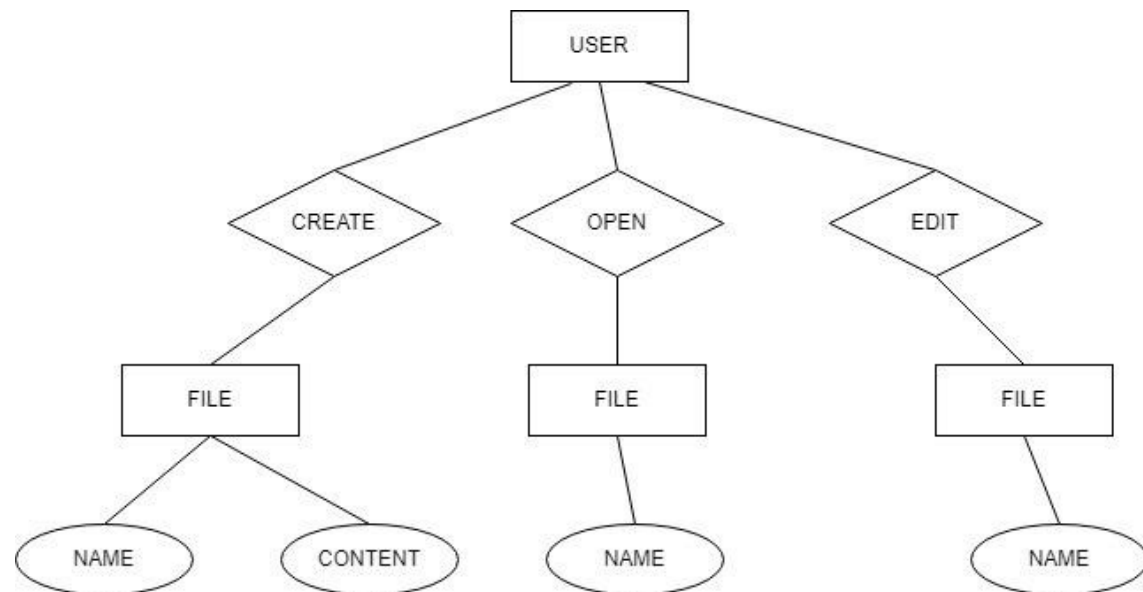
The editor is built using python and .config files are used to beautify the text editor . Required python inbuilt and other external modules will be used in the development.Finally heroku will be used to deploy the Text editor.

3.4 Communications Interfaces

The app will be deployed using Heroku so it can run on the local system or on any cloud based platform.

4. Analysis Models

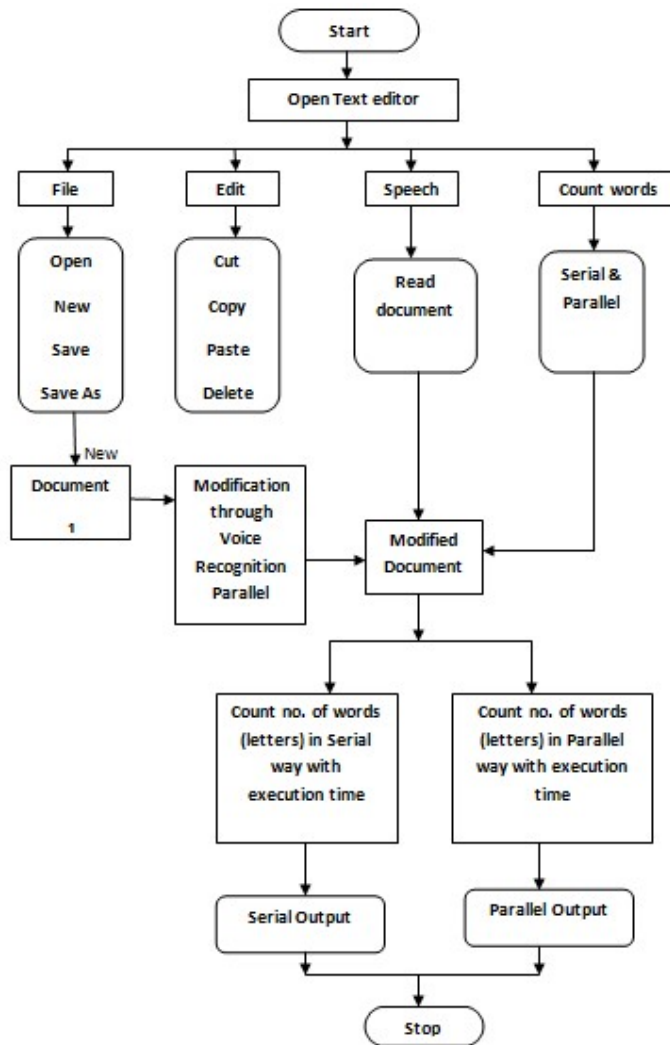
4.1 ER Diagram



4.2 UML



Data Flow Diagram for Text Editor

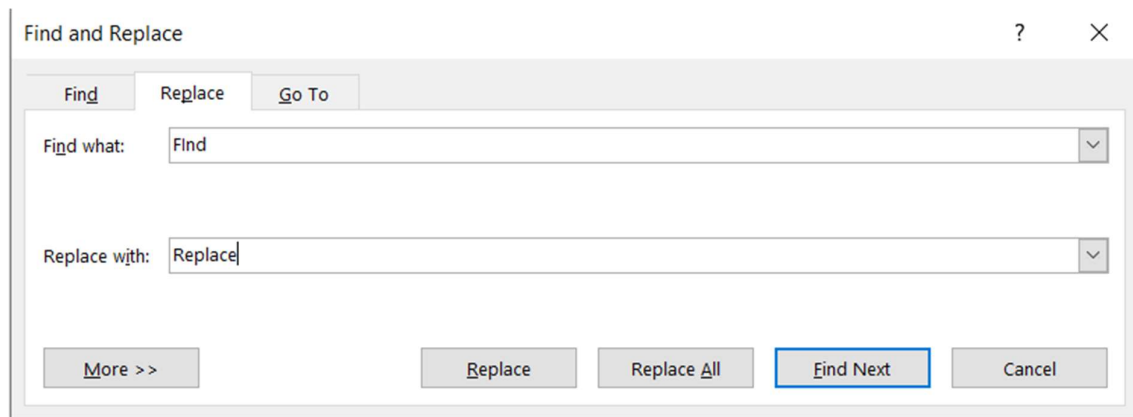


5. System Features

5.1.1 Features

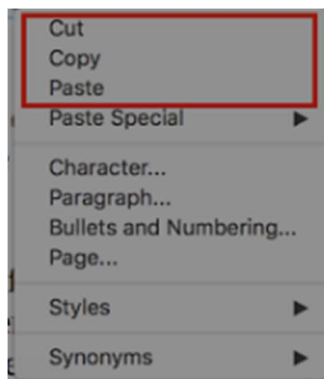
Feature 1) Find and Replace: Allows you to find all similar words and replace it with required word,

Priority: High



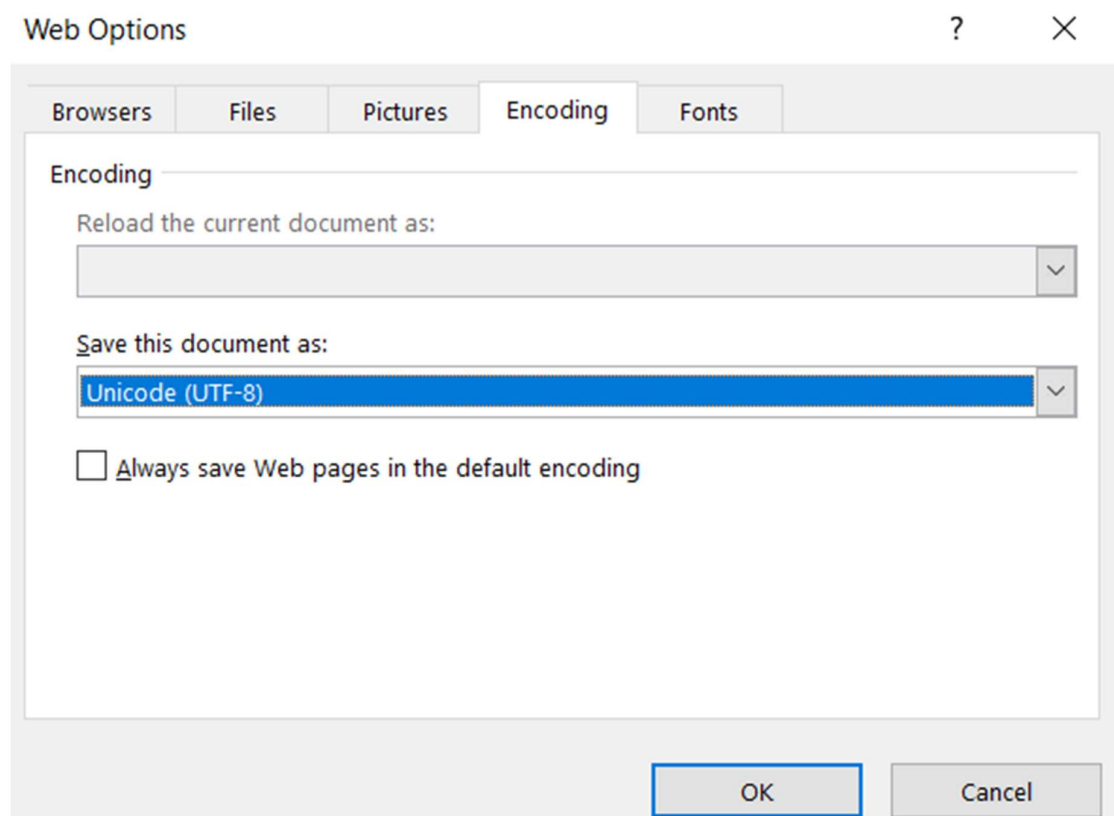
Feature 2) Cut, Copy and Paste: Allows the user to cut, copy and paste a word , text in the editor.

Priority: High



Feature 3) Ability to handle UTF-8 encoded text: Large text can be stored in UTF-8 encoding and can be brought back to string when needed

Priority: Medium



Feature 4)Text Formatting:Allows you to change style, size and color of text.

Priority:High



Feature 5)Undo and redo:Allows you to undo and redo on the editor

Priority:High



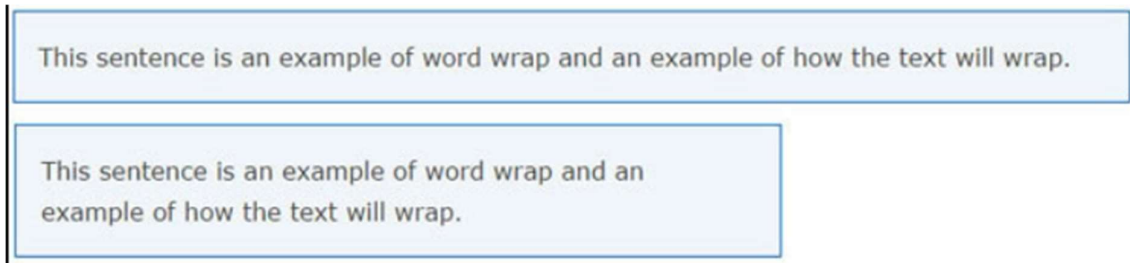
Undo



Redo

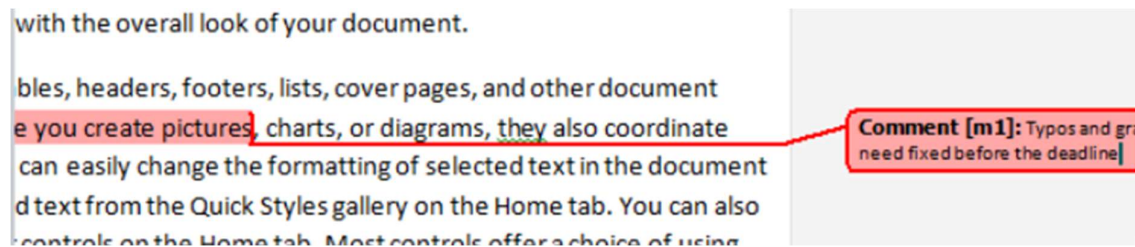
Feature 6)Line wrap:This user allows user to write text in multiple lines

Priority:High



Feature 7)Comment formatting:Enable comments in the editor

Priority : Medium



Feature 8)Syntax highlighting & Source Code Editor.

Priority : Medium

```
get area() {  
    return this.calcArea();  
}  
// Method  
calcArea() {  
    return this.height * this.width;  
}  
}  
  
const square = new Rectangle(10, 10);
```

5.1.2 Stimulus/Response Sequences

An user can be able to click on the various buttons, highlight a text, open a menu, scroll the screen, hover on some menus, etc...

5.1.3 Functional Requirements

They are :-

- Change User Preferences
- Develop Algorithm
- Use DGP Specific API
- Analyse Algorithm
- Debug Step by Step
- Use Visual Debugger
- Profile Algorithm
- Reference Output Comparison

6. Other Nonfunctional Requirements

6.1 Performance Requirements

Pauses, hangs and slow responses result in the tool interrupting the task at hand. Whenever possible, we should respond immediately to user input.

If the user closes the text editor with a file open after opening the text editor again it shows the file.

It has multiple interfaces, it shows normal text files and python code.

Separates the actual keyword from variable by making it colourful as shown below,

```
1  import pandas as pd
2  import numpy as np
3
4
5
6  def add_2(x, y):
7      return x + y
8
9
10
11 print(add(5, 6))
```

6.2 Safety and Security Requirements

Security risk in extensible text editors enables hackers to abuse plugins and escalate privileges.

This Editor helps to keep the code when the user closes the editor unexpectedly.

The plugins that we add to the text editor are made by our team and it will not be the third party plugins. Review 3rd party plugins code before approving their use in the network environment.

Folder permissions are to entirely separate the plugin folders that are used when running the editors in elevated models.

Deny write permissions for non-elevated users, by taking root ownership on the relevant plugins folder.

6.3 Software Quality Attributes

Usability of a user its ease, it based on principle KISS(Keep It Simple Stupid) and it is user friendly. As it can be used by a normal person as a text file or python code editor.

Reliability between user and software will be good because we are using Agile changes are accepted.

Portability of the software, it can be used in any operating system.

Scalability, the system will handle the demand for stress caused by increased usage without decreasing performance.

Flexibility, the software has the ability to change the future as we are using the Agile framework.

Additional Software Quality Attributes are Testability, Reusability, Maintainability, Supportability, Interoperability, Performance and Security.

6.4 Business Rules

The first release of the text editor will consist of a single user interface. Hence, a single user will have master control.

In the subsequent pipeline releases, we plan to release a multi-user interface for a document. This allows multiple users to create and edit a document or code in real-time.

In a multi-user interface, there are certain business rules that have to be implemented.

One of the main features is Read-Write access.

For example, in a Professor-Student or Author-Editor environment, we will provide limited access and functionality control to one user and master control to the other.

Upon expanding we can then classify each functionality based on user type and build a table. This will help to enforce customised business rules based on user requirements.

7. Other Requirements

7.1 Auto-Indenting

One of the most important features of a python programming environment is indentation. Tab/Space errors in python indentation will lead to errors.

Hence, it is an important customer requirement to add auto-indentation feature which will eliminate the possibility of customer physically indenting each line.

Version 1 of this feature will be available as a preview for programmers. Based on customer feedback and suggestions, this feature will be made available to all in later releases.

7.2 Auto-Saving

One of the other major customer requirements is that - in case of a sudden failure of the application or any other unforeseen circumstances, the data is auto-saved and backed up.

A cloud-based framework will be integrated in future releases to allow for auto-saving of the work done within a given Time-Based Refresh Rate.

This provides a back-up to the user to continue their work in case of crash/failure.

Appendix A: Glossary

UTF-8: A variable-width character encoding used for electronic communication. Defined by the Unicode Standard, the name is derived from Unicode (or Universal Coded Character Set) Transformation Format – 8-bit. UTF-8.

Line Wrap: is breaking a section of text into lines so that it will fit into the available width of a page, window or other display area.

Configuration Files: They are files used to configure the parameters and initial settings for some computer programs. They are used for user applications, server processes and operating system settings.

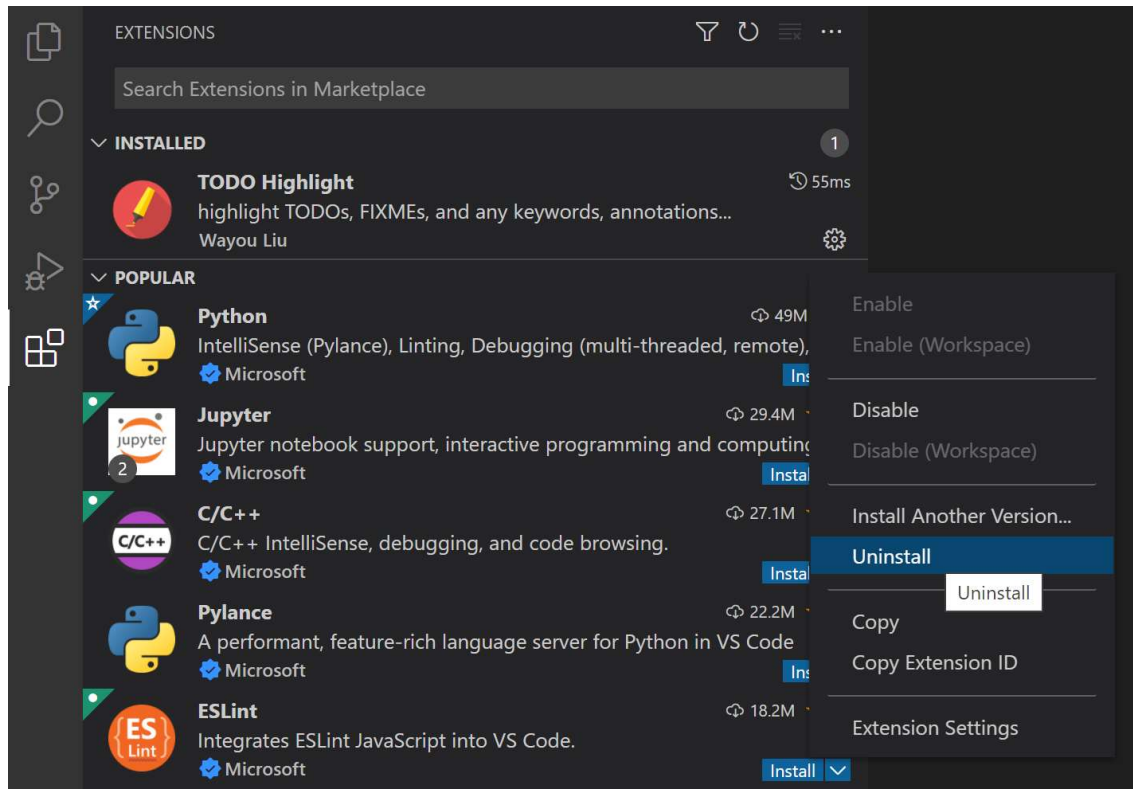
Heroku: A cloud-based development platform as a service (PaaS) provider. PaaS means Product as a Service.

Below is a list of definitions, acronyms and abbreviations required to interpret this document properly.

DGP	Digital Geometry Processing
CGI	Computer Generated Imagery
API	Application Programming Interface
IDE	Integrated Development Environment
GUI	Graphical User Interface

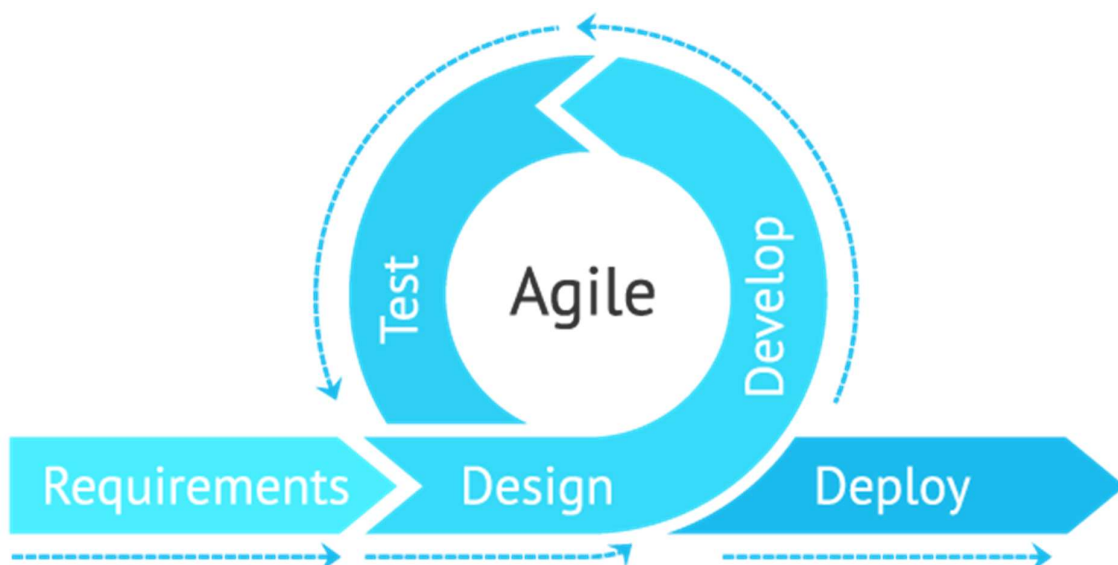
SRS	Software Requirement Specification
SDD	Software Design Document
SSD	Step by Step Debugger
WYSIWYI	What you see is what you implemented
EU	End User
UML	Unified Modelling Language
editor	Code Editor (Text Editor, are used interchangeably) (software component)
canvas	Interactive 3D Canvas (software component)
info	Info (software component)
header	Header (software component)
profiler	Profiler (software component)
debugger	Step by Step Debugger (software component)
output	Anything written to Info is output
mesh	A data structure in computer graphics
vertex	A data structure in computer graphics

Plugins or 3rd party plugins are things that are used to make the user's job easier. As shown in the following image:



Keep it simple, stupid (KISS) is a design principle which states that designs and/or systems should be as simple as possible.

An Agile framework is a specific approach to planning, managing, and executing work. The changes are welcome at any stage.



Appendix B: Field Layouts :-

As discussed in the above cases. Will produce a detailed explanation on the attributes on customer demands.

Appendix C: Requirement Traceability Matrix

Will be done, once testing phase starts.

Sl. No	Requirement ID	Brief Description of Requirement	Architecture Reference	Design Reference	Code File Reference	Test Case ID	System Test Case ID

TEAM CONTRIBUTION

TASK	NAME	SRN
SRS 3,4,5	ROHIT ROSHAN	PES1UG20CS355
SRS 1,2 Appendix A	S M SUTHARSAN RAJ	PES1UG20CS362
SRS 6.4, 7 Synopsis-Half	ROHAN C	PES1UG20CS345
SRS 5, 6, Appendix B Synopsis-Half	RAHUL ROSHAN G	PES1UG20CS320

THANK YOU FOR READING THIS
DOCUMENT PATIENTLY!!!