1. Introduction

For editing plain text files, a program called text editor is used. A plain text file is shown and edited by displaying all the characters the way they are involved in the file. UNICODE is the most frequently used character set because of more frequent use of plain text files for programming and configuration and less frequent use of them for documentation (for instance, detailed instructions, user guides) as compared with the past.

This system is designed keeping in mind the simplicity of notepad along with features like quick search, theme modifications like font style and colour customization, background. Quick search will reduce the struggle of searching meaning of words on the browser. It will provide all the features with a user-friendly interface.

1.1 Idea and Purpose

We suggest an app which will act as a common platform for easy, fast and effective text editing. This application will provide word search options within the application. It will provide simplicity of notepad along with features like quick in-app search, web search, and theme customization like changing style and colour font and background.

2. Methodology

2.1 Existing Methods

Presently, the following methods are used to make or edit text documents:

1. Heavy and slow document editors like MS word.
2. Notepad with no inbuilt high-level search functions.
3. Some text editors with no theme customization options.

2.2 Motivation

Create an application which provide simple, easy and quick editing of text documents, by improvising the existing systems using more modern and modular technology.

2.3 Suggested Solutions

We suggest the following solutions to the above-discussed problems:

1. Simple and user-friendly notepad like application.
2. Web search functionality.
3. Theme customization.

2.4 Feasibility Study

An Assessment of the feasibility of the project

Economic Feasibility

The project is economically feasible as it works with functions with low-cost services such as laptops and desktops.

Technical Feasibility

The current project is technically feasible as the application requires:

1. Any python supported IDE
2. GUI development tools

All these are readily available and can be successfully deployed on any available computer.

Behavioural Feasibility

The application is behaviourally feasible since it requires no technical guidance; all the modules are user friendly.

Operational Feasibility

The application is operationally feasible as:

1. Complete GUI-Base, which is user friendly.
2. Inputs to be taken are self-explanatory.
3. The system cuts down the load on client's machine.

2.5 Software Requirement Specifications

Software Requirement Specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements and may include a set of use cases that describe user interactions that the software must provide. It establishes the basis for an agreement between customers and the software providers on what the software product is to do and what it is not expected to do so that there is no room for confusion in the future. If used appropriately, SRS can help prevent software project failure.

Our proposed system has the following requirements:

1. The system requires storing the text documents on the host computer.
2. The system needs to provide web-based search functionality for words.
3. The system needs to update, delete, and modify the text document.
4. The system should be able to change the font style and colour of text.

3. Tools: Software and Hardware Requirements

3.1 Tools and Technologies

The following tools and technologies are expected to be used in development. Further may be added as the operations are implemented.

1. Language: Python
2. Libraries: Tkinter, wiki

3.2 Hardware Requirements

The following hardware requirements are recommended to be fulfilled in order to run this software.

1. CPU: Intel core i3 3rd Generation / AMD FX-6100
2. RAM: 1 GB
3. GPU: Integrated Graphics
4. Storage: 50 MB

3.3 Software Requirements

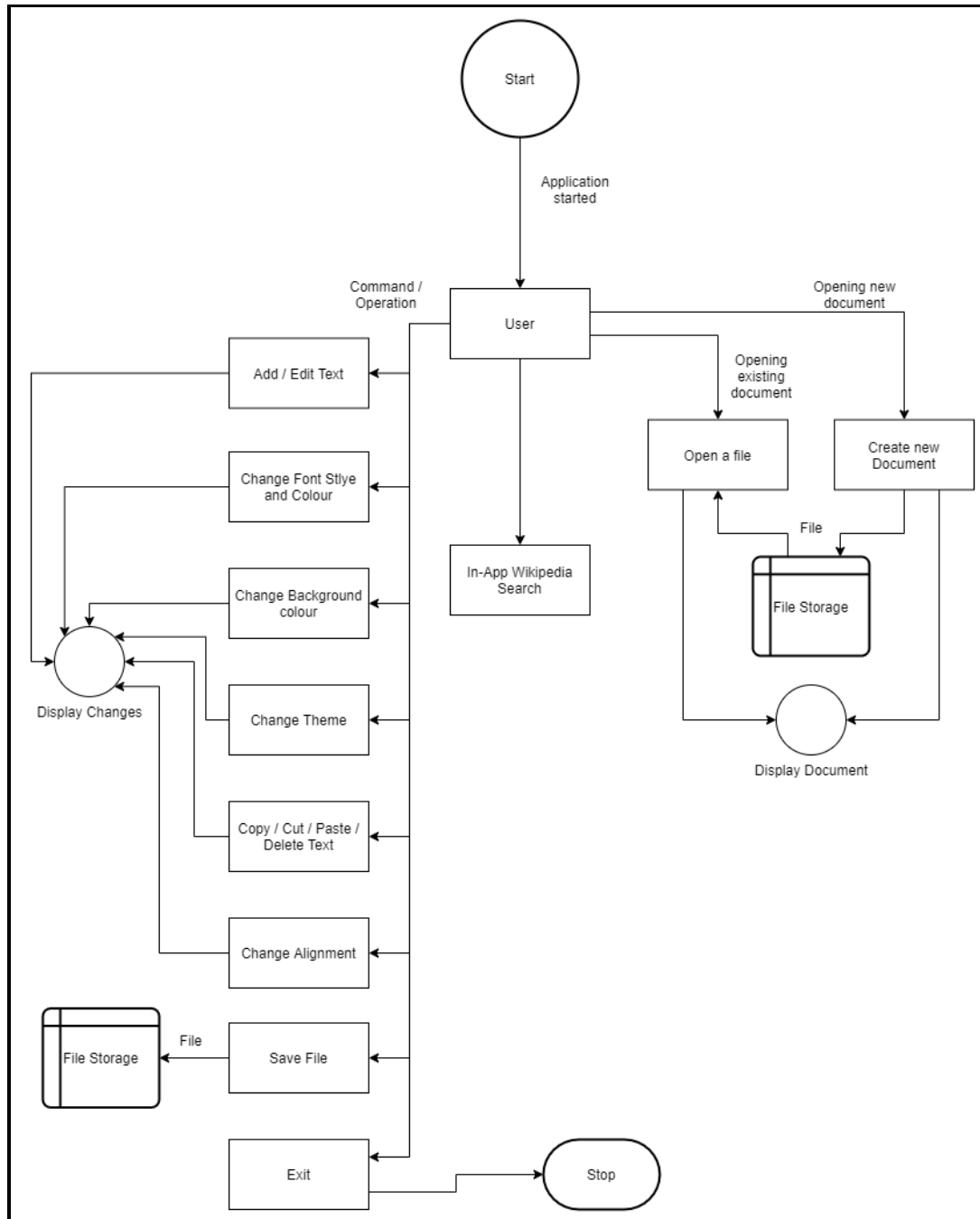The following software requirements are recommended to be fulfilled in order to run this software.

1. OS: Any Operating System
2. Programming Language: Python

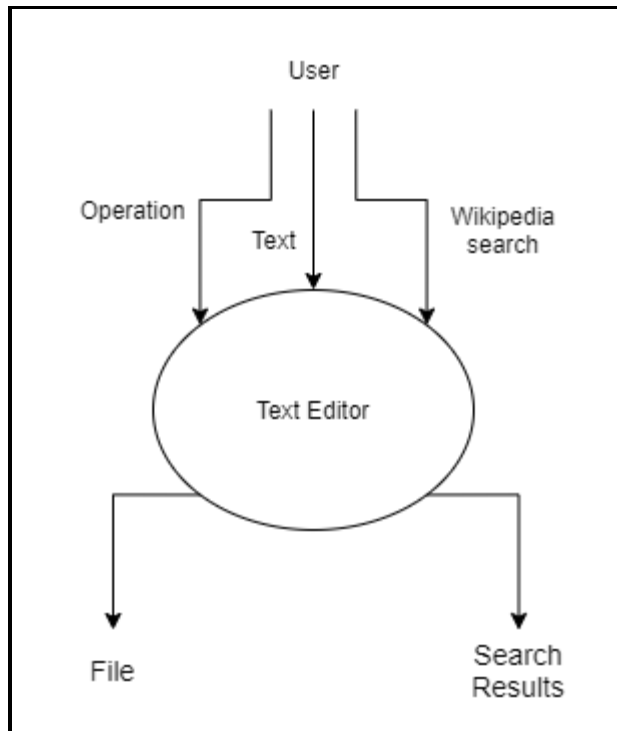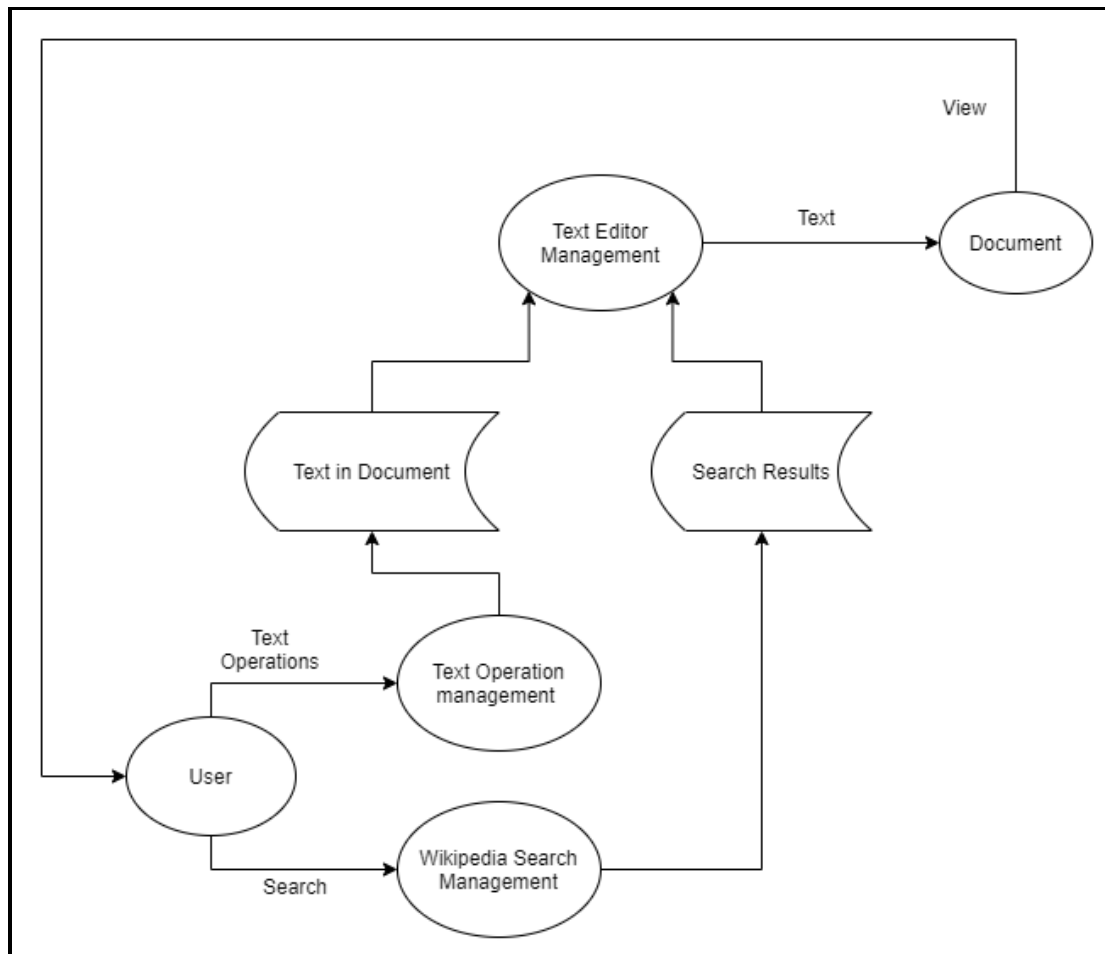## 4. System Design

4.1 Use Case Diagram:
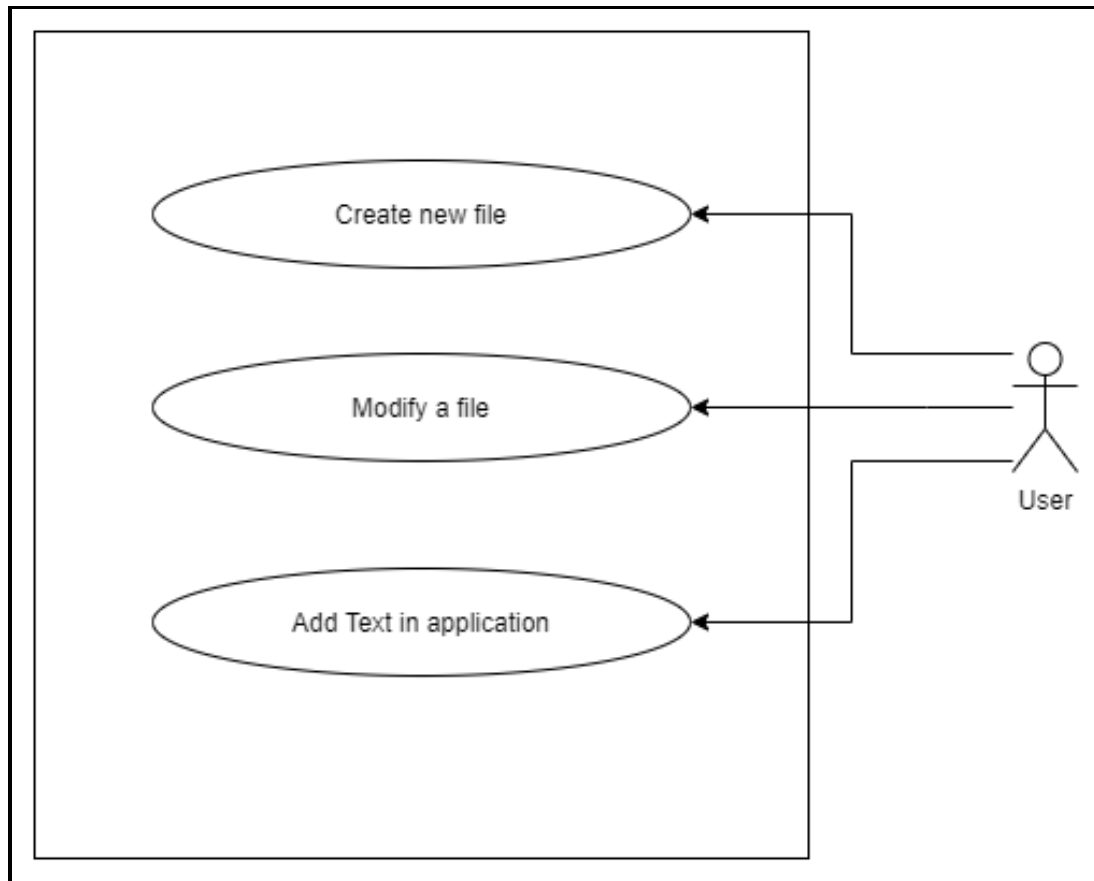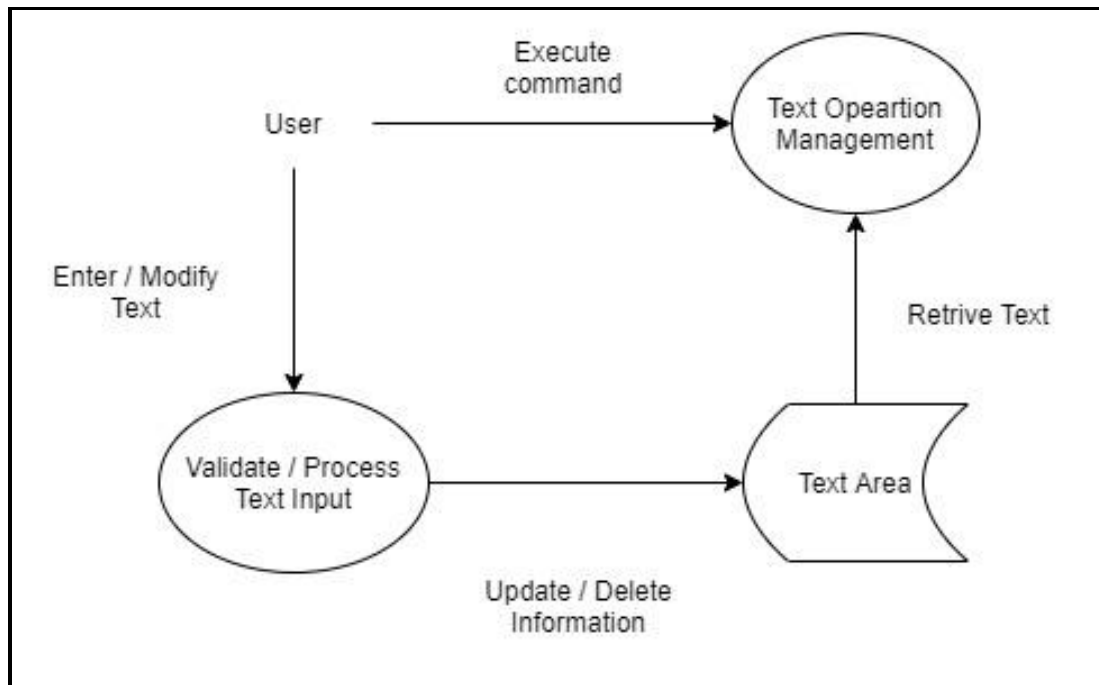
4.2 Flow Chart:

4.3 Data Flow Diagram (DFD):

Level 0:
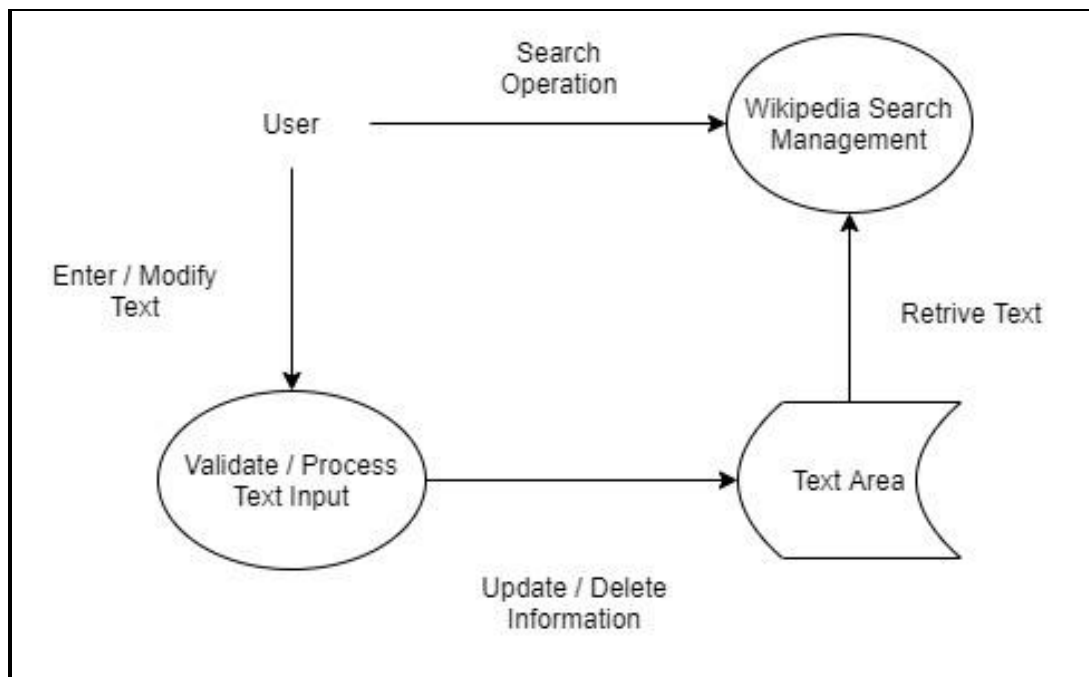
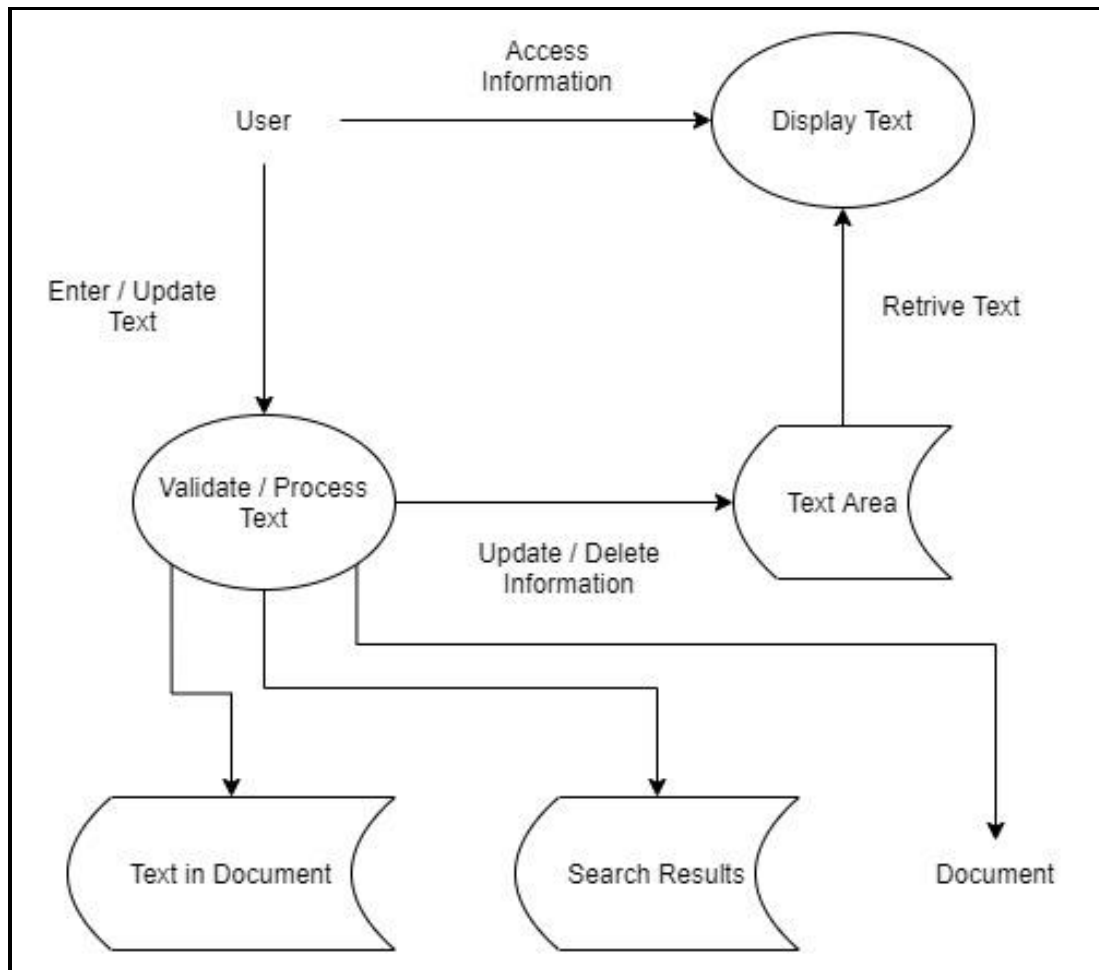**Level 1:**

Level 2:

Text Operation Management

Wikipedia Search Management

Text Editor Management

5. Future Scope

As of now this is a simple text editor that can be used to work on text-based files/documents. In the current pace of the world where technology is more than just relevant, so people learn coding & to do that they use IDE's like Visual Studio and PyCharm which are user friendly up to a limit, heavy and draining to the using system. Our Text editor can have plugins and packages that can be added by user to improve their experience, use the editor optimally for the programming language that they are learning.

6. Conclusion

The application has been developed keeping in mind the users of this generation who prefer to work on a flexible text editor that would provide a simpler and better hands on text or text editing experience than the existing notepad of windows whilst being almost the same size as it and will be future proof as the need for a reliable text editor is perpetual.