Simple File System Simulator App - Requirements Specification

1. Introduction

This document outlines the requirements for a Simple File System Simulator App, aiming to contribute to Sustainable Development Goal (SDG) targets related to Industry, Innovation and Infrastructure (SDG 9) by promoting education and awareness in computer science concepts.

2. Problem Statement

There is a need for educational tools that simplify complex concepts like file systems. A user-friendly file system simulator can bridge the gap between theoretical knowledge and practical application, fostering a better understanding of data storage and management.

3. Target Audience

Primary Audience: Students and individuals interested in learning about computer science fundamentals, particularly file systems.

Secondary Audience: Educators looking for interactive tools to supplement their curriculum in computer science or related fields.

4. Detail Requirements

4.1 Functional Requirements

4.1.1 CRUD Operations:

Users shall be able to Create, Read, Update, and Delete files and directories within the simulated file system.

4.1.2 Rudimentary Operations:

The application shall implement basic file system operations including:

Copy files and directories.

Move files and directories.

Rename files and directories.

Search for files within the simulated file system.

4.1.3 Visualization:

The application shall visually represent the simulated file system structure, allowing users to see directories and files in a hierarchical manner.

4.1.4 (Optional) Performance Monitoring:

(Optional) The application may provide insights into performance metrics like operation execution time or memory usage, allowing users to explore the impact of different file system operations.

4.2 Non-Functional Requirements

4.2.1 User Interface:

The application shall provide an intuitive and user-friendly interface that is easy to navigate and understand, catering to users with varying levels of technical expertise.

Consider offering interactive tutorials or help guides within the application.

4.2.2 Performance:

The application shall ensure smooth performance with minimal response times for user interactions.

5. Proposed Solution

A desktop application will be developed using Python as the primary programming language. Python offers a user-friendly syntax, extensive libraries for file system manipulation and user interface creation, and a large developer community for support. The application will simulate basic file system functionalities, allowing users to interact with a virtual file system environment and gain practical experience.

6. Tools and Language

6.1 Primary Tool: Python

6.2 Libraries (Potential):

os: Provides functionalities for interacting with the operating system's file system (for reference).

shutil: Offers utilities for file system operations like copying, moving, and renaming files and directories.

tkinter: A Python library for creating graphical user interfaces.

7. Future Considerations

Explore the possibility of incorporating gamification elements to enhance user engagement.

Consider developing a mobile application version to increase accessibility.

This document serves as a starting point for the development of the Simple File System Simulator App. Additional requirements may be identified and documented during the development process.