ABBOTTABAD UNIVERSITY OF SCIENCE & TECHNOLOGY

SOFTWARE REQUIREMENTS SPECIFICATION (SRS DOCUMENT)

For
File System Simulator
Version 1.0

By
Safiullah khan

Roll number 14795

BSCS 3rd D

Supervisor
(Sir Jamal Abdul Ahad)

TABLE OF CONTENTS

Introduction

Purpose

This Software Requirements Specification (SRS) document outlines the requirements for the File System Simulator project. It serves as a guide for developers, students, and instructors to understand the system's functionalities and use it effectively.

Document Conventions

- Bold Text: Indicates key terms or section headings.
- Italic Text: Provides notes or additional explanations.
- Numbered Lists: Shows steps or prioritized items.

Project Scope

Scope Definition

The File System Simulator is a learning tool designed to emulate the basic operations of a file system, such as creating, deleting, reading, and writing files. It is primarily aimed at providing practical exposure to file system concepts.

Core Features

- File Operations: Create, delete, read, and write files.
- Directory Management: Create and organize directories.
- File Allocation Methods: Simulate allocation techniques like contiguous, linked, and indexed allocation.
- Disk Management: Visualize and manage disk space with free space management.
- User Interface: A simple, easy-to-use interface with clear commands and options.

Subsequent Releases

Future versions may introduce:
- Advanced visualization tools for file and disk structures.
- Support for real-time collaboration or shared simulations.
- Extended support for different file system types.

Alignment with User and Business Goals

This project aligns with educational objectives by providing hands-on learning experience. It supports students and educators in bridging the gap between theoretical concepts and practical application.

References

- Operating System Concepts, Silberschatz et al.
- Relevant internal course materials and project guidelines.

Overall Description

Product Perspective

Product Context

The simulator is an independent application that does not require integration with external file systems, making it versatile and easy to use.

Product Origin

The project was initiated for understanding file system operations.

Product Relationship to Existing Systems

It operates as a standalone tool but can be extended to compare its results with real-world file systems.

Product Ecosystem

The product includes:
- A graphical or command-line user interface.
- A backend simulation engine for processing operations.
- Visualization components for illustrating file system structures.

User Classes and Characteristics

- Students: Learn and practice file system concepts through an interactive platform.
- Instructors: Use the tool to demonstrate theoretical concepts in a classroom setting.
- Researchers: Experiment with custom scenarios and evaluate file system behaviors.

Operating Environment

- Compatible with Windows and Linux operating systems.
- Requires at least 2 GB RAM and 500 MB free disk space.

Design and Implementation Constraints

- Limited complexity in simulations to ensure smooth performance.
- Developed using Python or Java, with simple and clear coding standards.

Assumptions and Dependencies

- Assumes basic knowledge of file systems among users.
- Relies on libraries for graphical interfaces and backend logic.

System Features

Refer to the table below:

Features	Description	Requirements
File Operations	Create, delete, read, and write files	User authentication
Directory Management	Organize files into directories	Hierarchical structure
File Allocation	Simulate allocation methods like contiguous, etc.	Efficient data

Disk Management	Visualize and manage disk space	Disk visualization
Reporting	Generate reports on file system usage	Data logging and analytics

External Interface Requirements

User Interfaces

- **Command-Line Interface**: Offers text-based operations with clear instructions.
- **Graphical User Interface**: Provides an intuitive, visual representation of the file system.

Software Interfaces

- Uses standard libraries for file handling, GUI design, and disk visualization.

Hardware Interfaces

- Runs on standard desktops or laptops with basic configurations.

Quality Attributes

Performance: Processes command in less than 2 seconds.
Reliability: Ensures data integrity and error-free operation.
Usability: Designed for easy navigation and interaction.
Security: Protects simulated data from corruption and unauthorized access.

Project Interface