Software Requirements Specification (SRS)
File Indexing Services

1. Introduction

1.1 Purpose

This document outlines the requirements for developing a file management system capable of indexing directories, searching for files, and organising file data efficiently.

1.2 Scope

The system facilitates directory indexing using various methods, including B-tree search and hashing. Additionally, it provides file search capabilities within indexed directories.

The system will be able to implement different DLLs for calling different services and will efficiently use different threads for different services.

2. Functional Requirements

2.1 Directory Indexing

2.1.1 Users can specify a directory path for indexing.

2.1.2 The system recursively traverses directories, indexing file names using B-tree search or hashing methods.

2.1.3 Indexed file data is stored in a data structure for efficient retrieval.

2.2 File Searching

2.2.1 Users can search for files within indexed directories by providing search strings.

2.2.2 The system utilizes file indexing data structures to perform efficient searches.

2.2.3 Search results are presented to the user, including file paths and relevant information.

3. Non-Functional Requirements

3.1 Performance

3.1.1 Indexing and searching operations should be efficient, even for directories with a large number of files.

3.1.2 The system should minimize response times for user queries, ensuring a smooth user experience.

3.2 Reliability

3.2.1 The system should handle errors gracefully, providing informative error messages to users.

3.2.2 Indexing and searching algorithms should be robust and resistant to data corruption.

3.3 Usability

3.3.1 The user interface should be intuitive and user-friendly, guiding users through indexing and searching processes.

3.3.2 Clear instructions and feedback should be provided to users during system operation.

4. External Interface Requirements

4.1 User Interface

4.1.1 The system may provide a command-line or graphical user interface (GUI) for user interaction.

4.1.2 Input forms and prompts should be clear and intuitive, facilitating user input.

4.2 Hardware Interface

4.2.1 The system should be compatible with standard hardware configurations, requiring minimal system resources.

4.2.2 Hardware requirements should be documented to ensure compatibility across different platforms.

5. Appendix

5.1 Glossary

- B-tree: A self-balancing tree data structure used for indexing and searching large datasets efficiently.
- Hashing: A technique for generating unique identifiers (hash codes) for data, facilitating efficient retrieval based on keys.

- Unicode: A computing standard for consistent encoding, representation, and handling of text expressed in most of the world's writing systems.
- File Indexing: The process of organizing and storing file data to enable fast and efficient retrieval based on search queries.