Project Overview

Topic: - File Management System

- Group Members
    1. Aditya Pratap Singh Chauhan (211154)
    2. Sushen Singh Rana (211135)
    3. Aditya Mittal (211146)

- Small Abstract of Project

1. Designing a hierarchical file structure to group files based on type or purpose.
2. Implementing file search and retrieval mechanisms, such as indexing and tagging.
3. Creating tools for file compression and encryption.
4. Integrating the file management system with other software, such as a database or a content management system.
5. Ensuring that only authorized users can access, modify or delete files through access control and permissions.
6. Implementing backup and disaster recovery strategies to protect files from accidental deletion or corruption.
7. Keeping track of different versions of a file and allowing users to easily revert to an earlier version if needed.
8. Synchronizing files across multiple devices or locations.
9. Allowing users to store and access files in the cloud for increased accessibility and collaboration.

Requirement Analysis

Topic: - File Management System

- Group Members
    1. Aditya Pratap Singh Chauhan (211154)
    2. Sushen Singh Rana (211135)
    3. Aditya Mittal (211146)

- Requirements
    1. User management: Ability to create, edit and delete user accounts, set user privileges and manage user access.
    2. File and folder management: Ability to create, edit, delete, copy, move and search files and folders, set file permissions and manage file versions.
    3. Storage management: Ability to manage and allocate storage space, track usage, monitor storage utilization and display storage statistics.
    4. Backup and recovery: Ability to schedule regular backups, restore files and folders, and ensure data integrity and security.
    5. Security: Implement robust security measures such as password protection, encryption, and access control to protect sensitive data and prevent unauthorized access.
    6. File sharing: Ability to share files and folders with specific users, set sharing permissions and manage shared files.
    7. User interface: A user-friendly interface that is intuitive and easy to use, with support for different operating systems and devices.
    8. Integration: Integration with other software and systems such as document management systems, email systems and cloud storage solutions.
    9. Scalability: The ability to scale the system to accommodate growing storage and user needs.
    10. Performance: The system should have high performance and response time, even when handling large files and a large number of concurrent users.
    11. Customization: The ability to customize the system to meet specific user or organizational needs, such as customizing the user interface, adding custom file types or adding custom workflow processes.

- Top Requirements
    1. File sharing: The ability to share files and folders with specific users and manage shared files is an easy priority as it will help users collaborate more effectively on projects and files.
    2. File and folder management: The ability to effectively manage and organize files and folders is essential. This includes creating, editing, deleting, copying, moving and searching files and folders, and setting file permissions.
    3. User management: Ability to create, edit and delete user accounts, set user privileges and manage user access.
    4. Storage management: Ability to manage and allocate storage space, track usage, monitor storage utilization and display storage statistics.

- Team member roles
    1. Aditya Pratap Singh Chauhan = File And folder management, Storage management:
    2. Aditya Mittal = User management, Storage management
    3. Sushen Singh Rana = File sharing, Storage management

Here is an elaboration of the software architecture for a file management system, including the identification of the classes (data, functions) for both the high-level design (HLD) and low-level design (LLD) documents:

High-level design (HLD):

The high-level design of the file management system will include the following components:

1. User Interface: This component will allow users to interact with the file management system by providing an easy-to-use interface. It will include the following functionalities:

Create a new file
Open an existing file
Save a file
Save a file as a different name
Delete a file
Rename a file
Copy a file
Move a file
Search for a file
Navigate through directories

2. Security Manager: This component will provide security to the files and directories. It will include the following functionalities:

Encryption and decryption of files
Authentication of users
Authorization of access to files

3. Database Manager: This component will manage the database that stores the file metadata. It will include the following functionalities:

Create a new database
Connect to an existing database
Store file metadata in the database
Retrieve file metadata from the database

Low-level design (LLD):

The low-level design of the file management system will elaborate on the high-level components and provide the identification of classes (data, functions) for each component:

1. User Interface:

Class: UserInterface
Data: N/A
Functions: createFile(), openFile(), saveFile(), saveAsFile(), deleteFile(), renameFile(), copyFile(), moveFile(), searchFile(), navigate()

2. File Manager:

Class: FileManager
Data:
File: filename, content, creation_date, modification_date, size
Directory: dirname, contents, creation_date, modification_date
Functions: createFile(), openFile(), saveFile(), saveAsFile(), deleteFile(), renameFile(), copyFile(), moveFile(), searchFile(), navigate()

3. Security Manager:

Class: SecurityManager
Data:
User: username, password, role
File: filename, encryption_key, access_list
Functions: authenticateUser(), authorizeAccess(), encryptFile(), decryptFile()

4. Database Manager:

Class: DatabaseManager
Data:
Metadata: filename, creation_date, modification_date, size, location
Functions: createDatabase(), connectDatabase(), storeMetadata(), retrieveMetadata()
Conclusion:

In conclusion, the file management system's software architecture includes four main components: User Interface, File Manager, Security Manager, and Database Manager. The identification of classes (data, functions) for each component has been elaborated in both high-level design (HLD) and low-level design (LLD) documents. This architecture provides an efficient and secure file management system for the operating system.