HLD/LLD:-
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.