**File Explorer Application**

**Project Title:** File Explorer Application

* **Objective:**

The objective of this capstone project is to develop a console-based file explorer application in C++ that interacts with the Linux operating system to perform essential file management tasks. The application will allow users to list files, navigate directories, manipulate files (create, copy, move, delete), search for files, and manage file permissions, all through a command-line interface.

* **Overview:**

The File Explorer application is designed to provide users with an intuitive and efficient way to manage their files and directories within a Linux environment. By leveraging the capabilities of the C++ Standard Library, particularly the <filesystem> module, this application handles various file operations with ease. It also includes logging functionality to keep track of user activities, which is critical for debugging and auditing purposes.

* **High-Level Design:**

The high-level design outlines the major components and their interactions.

**Main Program:**

Responsibilities: Handles user inputs, commands, and controls the overall flow of the application.

Components:

User Interface: Provides a command-line interface for user interactions.

Command Parser: Parses and interprets user commands.

**File Explorer Module**:

Responsibilities: Manages file and directory operations.

Components:

File Operations: Handles basic operations like listing, creating, copying, moving, and deleting files.

Directory Operations: Manages directory navigation and listing.

Search Functionality: Searches for files within directories.

Permission Management: Sets file permissions.

**Logger Module:**

Responsibilities: Logs important events and actions to a file.

Components:

Log Writer: Writes log messages to a file with timestamps.

* **Low-Level Design:**

The low-level design provides detailed descriptions of each module's classes, functions, and data flow.

**FileExplorer Class:**

Methods:

listFiles (const std::string& directory): Lists all files in the specified directory.

changeDirectory (const std::string& directory): Changes the current working directory.

searchFiles (const std::string& directory, const std::string& filename): Searches for a file within the directory.

setPermissions (const std::string& filename, int mode): Sets permissions for the specified file.

createFile(const std::string& filename): Creates a new file.

copyFile (const std::string& source, const std::string& destination): Copies a file from the source to the destination.

moveFile (const std::string& source, const std::string& destination): Moves a file from the source to the destination.

deleteFile (const std::string& filename): Deletes the specified file.

**Logger Class:**

Methods:

Logger(const std::string& filename): Constructor that opens a log file.

~Logger(): Destructor that closes the log file.

log(const std::string& message): Logs a message with a timestamp.

**Main Function:**

Logic:

Initialize Logger and FileExplorer objects.

Run a loop to receive and process user commands.

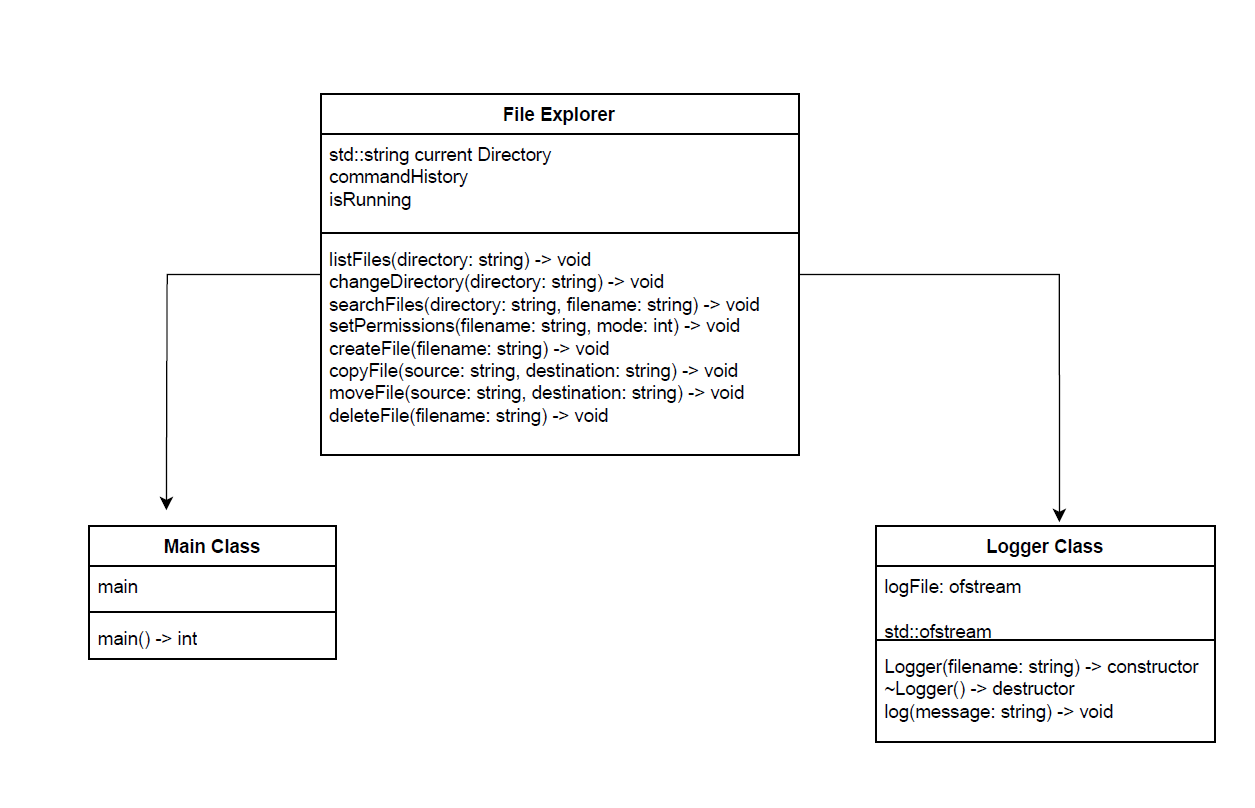
Call appropriate methods in the FileExplorer class based on user input

Log actions using the Logger class

**Activity Diagram:**

****

**Class Diagram:**

****