Platform: .Net6, C#

Create a project that has a Class Library and a Console app. We will implement most of the functionality below in the class library. The console app will use the class library and perform the user facing input/output required.

When performing the following tasks, name the variables appropriately, design the code to be modular (split into classes and methods as appropriate), and use efficient data structures.

1. In the class library, use the file system watcher class (https://docs.microsoft.com/en-us/dotnet/api/system.io.filesystemwatcher?view=net-6.0) to detect file changes in a given folder, such "Documents". Report the full file path to console on the following events:
    a. New file created
    b. File moved
    c. File changed
    d. File deleted
2. Determine if we get file changed events while a file is open in another program (say, a file is opened in Word and the user saves some changes but does not close the file), or only when the file is closed by the other program. Based on that determine a strategy to detect when a changed file is closed and is available for our app to open or process.
    a. There is no direct event for detecting file being closed, this but an idea is proposed on the following two links: https://bytes.com/topic/c-sharp/answers/265821-filesystemwatcher-detecting-file-closing-how and https://www.intertech.com/avoiding-file-concurrency-using-system-io-filesystemwatcher/
    b. Design your strategy after reviewing the above ideas. Analyze the expected delay in detection for your design. How can we determine the list of files to observe for which we need to check when the file is closed? What happens if the user saves an open file to a new file name (such as using the "Save As" menu option in many programs)?
3. Suppose we want to query the list of files in the entire Documents folder (a folder with a large number of subfolders and files) to find the list of files that have been modified within a given past duration (say the last one hour). Example: The program is started at 9am and we want to detect all files that changed within 8am to now. Write code for the following:
    a. Determine the list of files changed within the last given duration (say, last hour).
    b. Report the time it took to execute and the total number of files and folders in the directory for which the above query was performed.
    c. You may use the method below as appropriate, or other .Net6 methods:

        i. https://docs.microsoft.com/en-us/dotnet/api/system.io.fileinfo?view=net-6.0,

1. **Test:** Write an integration test (using xUnit, or another test framework of your choice) that tests the method above to query the list of modified files. (The test should not mock the file system but use the file system with file changes created by the test fixture.)
2. **Advanced Task:** Repeat query for files changed: After the information has been obtained once, if we want to update the list again (say, after an hour), are there any optimizations that could be used to make the next listing more efficient. Write a method that uses any information saved from the first query to perform the next query. What data structure would you use to save the results of the first run, so that the results can be used for the next run?
    a. You may use the methods below as appropriate, or other .Net6 methods:
        i. https://docs.microsoft.com/en-us/dotnet/api/system.io.fileinfo?view=net-6.0,
        ii. https://docs.microsoft.com/en-us/dotnet/api/system.io.directory.getfilesystementries?view=net-6.0 and
        iii. https://docs.microsoft.com/en-us/dotnet/api/system.io.directory.getlastaccesstimeutc?view=net-6.0 (Does accessing a file in a directory change the last access time of a directory?)
    b. If this task cannot be included in 10 hours of work, skip it.