Sprints Planned and Tasks Achieved:

Sprint 1 (Duration: 5 working days):

Set up the development environment (Eclipse/IntelliJ, Java, Git, GitHub).
Create the project structure and initialize the Git repository.
Implement the welcome screen with application name and developer details.
Display user interface options for interaction.

Sprint 2 (Duration: 5 working days):

Implement the first option to display file names in ascending order.
Handle scenarios where the root directory is empty or contains files/folders.

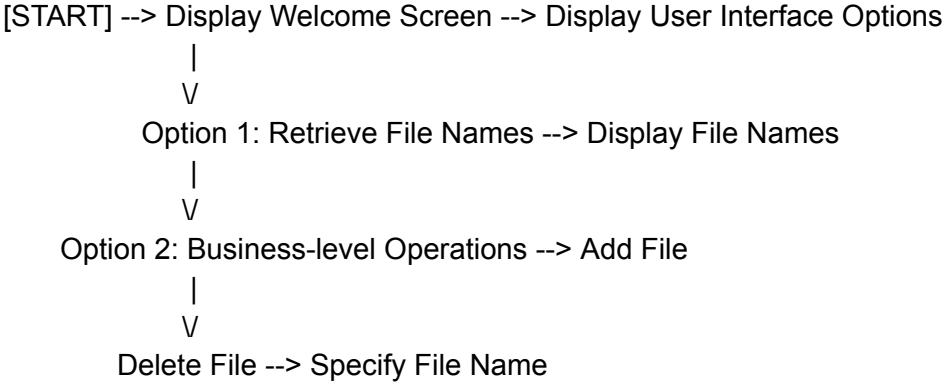
Sprint 3 (Duration: 5 working days):

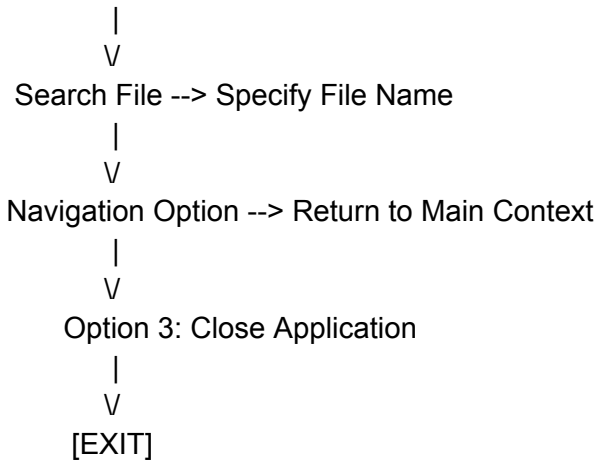
Implement the second option for business-level operations.
Add functionality to add a user-specified file to the application.
Add functionality to delete a user-specified file from the application.
Add functionality to search a user-specified file from the application.
Add navigation option to return to the main context.

Sprint 4 (Duration: 2 working days):

Implement the third option to close the application.
Perform code optimization and ensure exception handling.
Prepare for final release.

Flowchart:





Algorithm:

1.Algorithm for retrieving file names in ascending order:

Retrieve the list of files in the root directory.
Sort the list of file names in ascending order.
Display the sorted list of file names.

2.Algorithm for adding a file:

Prompt the user to enter the name of the file to be added.
Check if a file with the same name already exists.
If yes, display a message indicating that the file is already present.
If no, create a new file with the specified name in the root directory.
Write the default content ("The file contains confidential information.") to the file.
Display a message indicating that the file has been successfully added.

3.Algorithm for deleting a file:

Prompt the user to enter the name of the file to be deleted.
Check if a file with the specified name exists.
If yes, delete the file from the root directory.
Display a message indicating that the file has been successfully deleted.
If no, display a message indicating that the file is not found.

4.Algorithm for searching a file:

Prompt the user to enter the name of the file to be searched.
Check if a file with the specified name exists.
If yes, display a message indicating that the file is present.

If no, display a message indicating that the file is not found.

Core Concepts Used in the Project:

File I/O: Used to interact with files in the file system, such as reading file names, creating files, and deleting files.
Exception Handling: Implemented to handle potential errors and exceptions during file operations and user interactions.
Collections: Utilized to store and manipulate the list of file names.
Sorting Techniques: Applied to sort the list of file names in ascending order.
String Manipulation: Used to compare file names and perform case sensitivity checks.