Sprint planning

The following project is planned in one sprint-

1)Designing application
*Creating a class(filepath.java) to pass the path of directory
*Creating a class(operationonfile.java) to perform the following operation on given file -

1)Listing all the files in the given directory.
2)Add the files in the given directory.
3)Search the files in the given directory.
4)Deleting the files in the given directory.

*Flow chart of the application
2)Pushing the code to github repositories-**https://github.com/utkarsh136/Phase1-project-Virtual-key-of--repository**

Core concepts used in application

1)File handling
2)Collections
3)Sorting
4)Switch case
5)Exception handling

User stories

1)public filepath(user will give the path of his file)
2)Operations on file

```
while (true) {
                switch (r) {
                case 1: {
                        System.out.println("Listing  files in the  your given directory...!!!");
                        Set<String> l1 = displayFiles(your_filepath);
                        Iterator itr = l1.iterator();
                        while (itr.hasNext()) {
                                System.out.println(itr.next());
                        }
```

```
                    break;

                }

                case 2: {

                    System.out.println("Enter the file name that you
want to add.");

                    sc.nextLine();
                    String fName = sc.nextLine();
                    String escaped = your_filepath.replace("\\",
"\\\\");
                    addFile(escaped + "\\" + fName);
                    System.out.println("Your file got successfully
added.");
                    break;
                }
                case 3: {
                    System.out.println("Enter the fileName you want
to search.");
                    sc.nextLine();
                    String fName = sc.nextLine();
                    Set<String> l1 = displayFiles(your_filepath);

                    checkFileAvailability(l1, fName);
                    break;
                }
                case 4: {
                    System.out.println("Enter the file name that you
want to delete.");
                    sc.nextLine();
                    String escaped = your_filepath.replace("\\",
"\\\\");
                    String fName = sc.nextLine();
```

```
                    deleteFiles(your_filepath + "\\" + fName);

                    break;
                }
                case 5: {
                    System.out.println("Thank You..!!");
                    return;
```

3)Operation functions

```
    private static Set<String> displayFiles(String your_filepath) {
        File file = new File(your_filepath);
        String[] fileList = file.list();

        List<String> l = Arrays.asList(fileList);
        Collections.sort(l);

        Set<String> h = new TreeSet<String>();
        Iterator it = l.iterator();
        while (it.hasNext()) {
            h.add((String) it.next());
        }
        return h;

    }

    private static void checkFileAvailability(Set<String> hl, String
fName) {

        if (hl.contains(fName)) {
            System.out.println("File Found");
        } else {
            System.out.println("Not Found");
        }

    }
```

```
    private static void deleteFiles(String your_filepath) {

        try {
            Files.deleteIfExists(Paths.get(your_filepath));
        } catch (NoSuchFileException e) {
            System.out.println("No such file/directory exists");
        } catch (DirectoryNotEmptyException e) {
            System.out.println("Directory is not empty.");
        } catch (IOException e) {

        }

        System.out.println("Deletion successful.");
    }

}
```

Used the following:

- Eclipse/IntelliJ: An IDE to code for the application
- Java: A programming language to develop the prototype
- Git: To connect and push files from the local system to GitHub
- GitHub: To store the application code and track its versions
- Scrum: An efficient agile framework to deliver the product incrementally
- Search and Sort techniques: Data structures used for the project
- Specification document: Any open-source document or Google Docs