PHASE 1 ASSIGNMENT

FILE HANDLING SYSTEM

Developer: RACHEL PHILLIP

I'm a Vision Application Engineer. I design camera-based applications for inspection systems. The applications we design are PC based, and hence this course to get familiarised with web development.

TABLE OF CONTENTS

1. SPRINTS PLANNING 2
2. FLOWCHART OF THE APPLICATION 4
3. ALGORITHM 5
4. CORE CONCEPTS USED FOR THIS PROJECT 6
5. CONCLUSION 7

SPRINTS PLANNING

The problem statement required me to create a Prototype for LockedMe.com in **3 Weeks**.

Thus, I planned 3 Sprints with each Sprint having a duration of 1 week.

We had a Sprint Planning Meeting each week for a duration of 2 hours.

SPRINT 1:

Tasks Planned:

1. Plan the Features of the Application
2. Decide on the best Data Structure that should be used for saving the Files, to make sure Adding/Deleting Data is efficient, searching an element is quick, and Sorting of the array is fast.

Tasks Achieved at the End of the Sprint:

1. All features in the application were decided.
2. **Tree Set** was decided as the Data Structure which seemed the most precise for this application

SPRINT 2:

Tasks Planned:

1. Write a class for the File Handling Features, i.e., to Add, Delete, and Search the file. It also includes handling all exceptions, and displaying the respective Successful or Unsuccessful Message.
2. Write a Main Class to Handle the Inputs entered by the user, and for the User Interface to be displayed on the console.

Tasks Achieved at the End of the Sprint:

Both tasks were completed individually, but both classes needed to be integrated.

SPRINT 3:

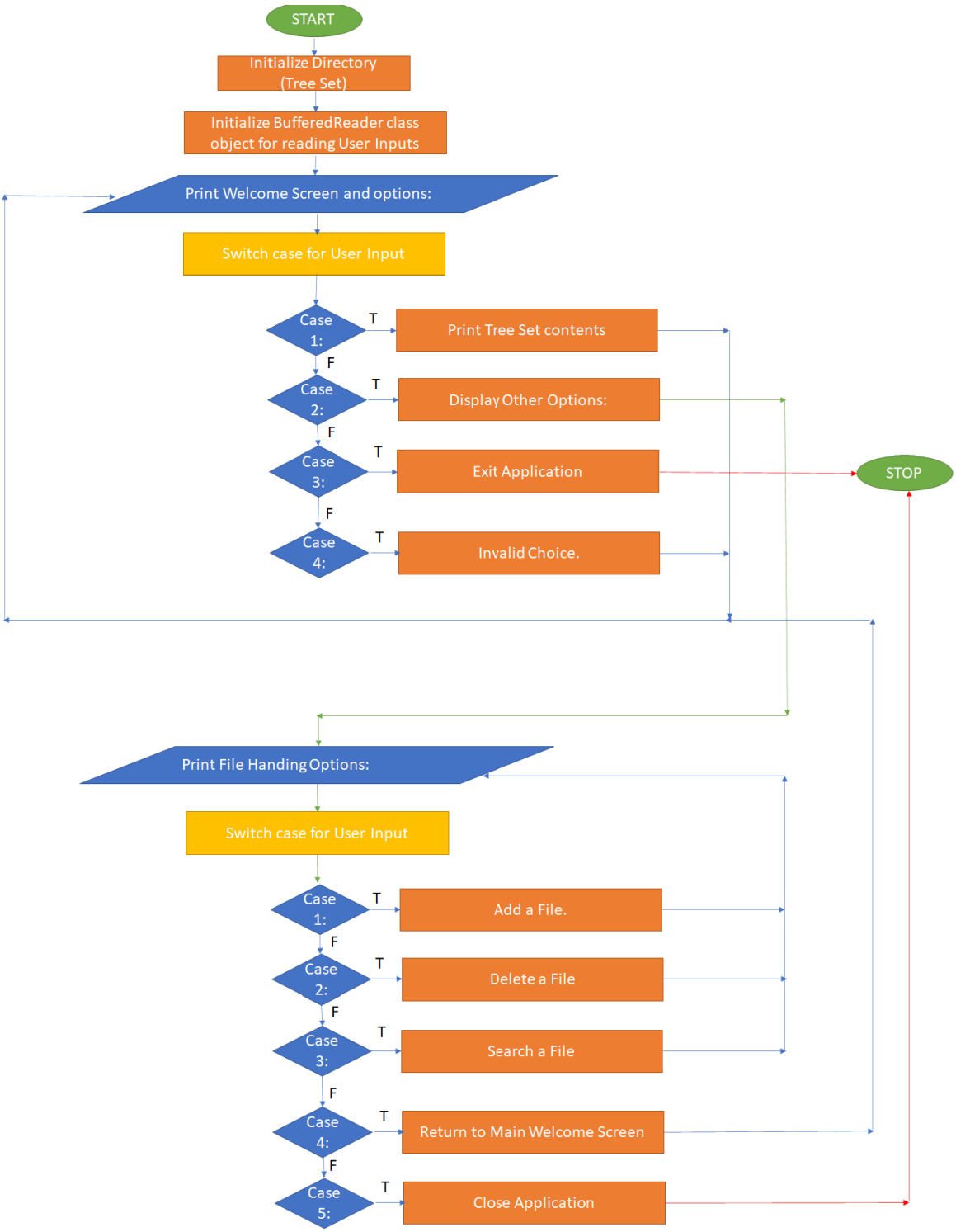
Tasks Planned:

1. Integrate both classes developed in Sprint 2.
2. Complete testing of the application
3. Prepare documentation for the application.

Tasks Achieved at the End of the Sprint:

All the tasks were successfully completed within the stipulated time.

FLOWCHART


START
Initialize Directory (Tree Set)
Initialize BufferedReader class object for reading User Inputs
Print Welcome Screen and options:
Switch case for User Input
Case 1:
T
Print Tree Set contents
F
Case 2:
T
Display Other Options:
F
Case 3:
T
Exit Application
F
Case 4:
T
Invalid Choice.
STOP
Print File Handing Options:
Switch case for User Input
Case 1:
T
Add a File.
F
Case 2:
T
Delete a File
F
Case 3:
T
Search a File
F
Case 4:
T
Return to Main Welcome Screen
F
Case 5:
T
Close Application

ALGORITHM

The requirement of the application was to have a data structure which can Add and Remove data easily, and also Sort and Search elements with ease.

Since the data structure we chose is a Tree Set, all these properties are inherent.

Tree Set is sorted Alphabetically by Default, when a string Tree set is used. Thus, making Sorting the Files easy.

Tree Sets allows easy Insertion and Removal of the elements, and hence different methods were used to adding, deleting, searching, and printing the Files in the directory.

On the Welcome Screen, we displayed 3 Options:

1. Print Files int the current directory in Ascending Order
2. Display other Options
3. Close Application

Unless the “Close Application” option is selected, the options are displayed in a loop.

When option 2 is clicked, the following Options were displayed:

1. Add a File to the Current Directory
2. Delete a File from the Current Directory
3. Search a File from the Current Directory
4. Go Back to Main Page
5. Close Application

CORE CONCEPTS USED

The following are the key concepts used in this Project:

1. **Data Structures**: I analysed all the Data Structures, to zero in on Tree Set for this Application. Thus, learning how big a part the data structure plays in making the application efficient.
2. **File Handling:** The concept of File Handling to take Inputs from the user. BufferedReader and InputStreamReader class objects were used to read the inputs entered by the user.
3. **Static & Non-Static Functions and Variables**: The concept of static and non-static functions was used to decide which functions must be static and which must be non-static.
4. **Git Concepts**: GitHub is used as the Version Control software for maintaining the versions of the software.
5. **Agile Scrum**: The agile scrum concept of Project Management was implemented to complete this Project.

CONCLUSION

This application has a potential to become a File Management System.

This prototype uses a Tree Set of Strings, just to Manage the files. But in the final software, we can use Tree with Nodes as Directories and at each Node, have Sub Nodes as Directories or Files.

Files of different formats can be saved, and we can try to filter the files based on the type, e.g. .xlsx, .txt, etc.

The USPs for this application is that there is no limit on the number of files that can be saved. And the sorting/ searching will be efficient even for a large set of Files, since Tree Set is used.