**1. Objective**

**Project objective:**

As a Full Stack Developer, complete the features of the application by planning the development in terms of sprints and then push the source code to the GitHub repository. As this is a prototyped application, the user interaction will be via a command line.

**Background of the problem statement:**

Company Lockers Pvt. Ltd. hired you as a Full Stack Developer. They aim to digitize their products and chose LockedMe.com as their first project to start with. You’re asked to develop a prototype of the application. The prototype of the application will be then presented to the relevant stakeholders for the budget approval. Your manager has set up a meeting where you’re asked to present the following in the next 15 working days (3 weeks):

* Specification document - Product’s capabilities, appearance, and user interactions
* Number and duration of sprints required
* Setting up Git and GitHub account to store and track your enhancements of the prototype
* Java concepts being used in the project
* Data Structures where sorting and searching techniques are used.
* Generic features and three operations:
  + Retrieving the file names in an ascending order
  + Business-level operations:
    - Option to add a user specified file to the application
    - Option to delete a user specified file from the application
    - Option to search a user specified file from the application
    - Navigation option to close the current execution context and return to the main context
  + Option to close the application

The goal of the company is to deliver a high-end quality product as early as possible. 

**The flow and features of the application:**

* Plan more than two sprints to complete the application
* Document the flow of the application and prepare a flow chart
* List the core concepts and algorithms being used to complete this application
* Code to display the welcome screen. It should display:
  + Application name and the developer details
  + The details of the user interface such as options displaying the user interaction information
  + Features to accept the user input to select one of the options listed
* The first option should return the current file names in ascending order. The root directory can be either empty or contain few files or folders in it
* The second option should return the details of the user interface such as options displaying the following:
  + Add a file to the existing directory list
    - You can ignore the case sensitivity of the file names
  + Delete a user specified file from the existing directory list
    - You can add the case sensitivity on the file name in order to ensure that the right file is deleted from the directory list
    - Return a message if FNF (File not found)
  + Search a user specified file from the main directory
    - You can add the case sensitivity on the file name to retrieve the correct file
    - Display the result upon successful operation
    - Display the result upon unsuccessful operation
  + Option to navigate back to the main context
* There should be a third option to close the application
* Implement the appropriate concepts such as exceptions, collections, and sorting techniques for source code optimization and increased performance

**You must use 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

**Following requirements should be met:**

* The source code should be pushed to your GitHub repository. You need to document the steps and write the algorithms in it.
* The submission of your GitHub repository link is mandatory. In order to track your task, you need to share the link of the repository. You can add a section in your document.
* Document the step-by-step process starting from sprint planning to the product release.
* Application should not close, exit, or throw an exception if the user specifies an invalid input.
* You need to submit the final specification document which includes:
  + Project and developer details
  + Sprints planned and the tasks achieved in them
  + Algorithms and flowcharts of the application
  + Core concepts used in the project
  + Links to the GitHub repository to verify the project completion
  + Your conclusion on enhancing the application and defining the USPs (Unique Selling Points)

**2. Sprints**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Date | 26/02 | 27/02 | 31/02 | 01/03 | 02/03 | 03/03 | 05/03 | 06/03 | 07/03 | 08/03 |
| Task | Planning and Scoping | Code Template and Workflow | Coding Menu part, loops & List File Function | Coding Menu part, loops & List File Function | Coding Business Operation, loops | Coding Add, Delete, & Search functions | Testing and Improvements in Code Logic Flow | Testing and Improvements in Code Logic Flow | Bug Fixes and Improvement | Documentation |

**3. Repository**

GitHub Link: https://github.com/smarthardik10/LockersMe-Java-Project

**4. Concepts**

Create a Folder before Running the programme in such a way – “C:\My\_Files”

* 1. Class - A class is a blueprint or prototype that is defined by the user and used to create objects. It denotes the set of properties or methods that all objects of the same type share.
  2. Scanner - The Scanner class in the java.util package is used to obtain input of primitive types such as int, double, and strings.
  3. If-else statement - The condition is tested using the Java if statement. It determines whether a boolean condition is true or false. In Java, there are several types of if statements.
  4. do while loop - The Java do-while loop is used to repeatedly iterate a section of a programme until the specified condition is met. A do-while loop is recommended if the number of iterations is not fixed and the loop must be executed at least once.
  5. Java.io.Files - The File class represents a file or directory pathname in Java. A simple string is insufficient to name files and directories because they have different formats on different platforms. Working with the pathname, deleting and renaming files, creating new directories, listing the contents of a directory, and determining several common attributes of files and directories are all covered by the File class.
  6. For loop - We can set the variable's value, check the condition, and increment or decrement the value. It is divided into four sections:

Initialization: When the loop begins, the initial condition is executed once. We can either initialise the variable or use one that has already been initialised. It's a condition that can be turned off.

Condition: This is the second condition, which is executed every time the loop is tested. It keeps running until the condition is false. It must return either true or false as a boolean value. It's a condition that can be turned off.

Increment/Decrement: It increases or decreases the value of the variable. It's a condition that can be turned off.

Statement: The loop's statement is executed each time the second condition is false.

* 1. Try catch – When running Java code, a variety of errors can occur, including programmer errors, errors caused by incorrect input, and other unforeseeable events. When an error occurs, Java usually comes to a halt and displays an error message. Java will throw an exception is the technical term for this (throw an error).
  2. Arrays - Depending on the array's definition, it can contain primitives (int, char, etc.) as well as object (or non-primitive) references to a class. The actual values of primitive data types are stored in contiguous memory locations. The actual objects are stored in a heap segment in the case of class objects. There are two versions of the Arrays.sort() method. one in which we don't pass any arguments and it sorts down the entire array, whether it's an integer or a character array; however, if we want to sort a specific part of the array using this method of the Arrays class, we overload it and pass the array's starting and ending indexes.
  3. Java.io.IOException - The most common cause of an IOException is when the user enters incorrect data into the programme. This could be the name of a file that doesn't exist or data types that the programme can't handle. When this happens, an exception (IOException) is thrown, informing the compiler that there was invalid input or output.

**5. Screen Shots**

**A computer screen capture

Description automatically generated with medium confidenceA computer screen capture

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidenceA computer screen capture

Description automatically generated with medium confidenceA computer screen capture

Description automatically generated with medium confidenceA picture containing text, monitor, screenshot, screen

Description automatically generatedA computer screen capture

Description automatically generated with medium confidenceA picture containing text, screenshot, monitor, electronics

Description automatically generated**