INDEX

|  |  |
| --- | --- |
| **Contents** | **Page No** |
| Specification Document | 2 |
| Number and duration of sprints | 4 |
| Flow of application | 5 |
| Core java concepts used | 7 |
| Data structure and algorithm used  Conclusion on enhancing the application | 8  9 |
| GitHub repository link | 10 |

**Specification Document:**

Product Name: LockedMe.com

Product description: A prototyped application which does various file operations.

As this is a prototyped application, the user interaction will be via a command line and actual file handling is not applied here. We are taking file as user input in the form of strings and doing various operations like add, delete, search, sort, display etc, using some data structures (ArrayList).

Developer: Madhav Bhat K

## Product’s capabilities:

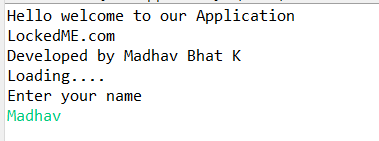
* Retrieving the file names in an ascending order
* Add a user specified file to the application
* Delete a user specified file from the application
* Search a user specified file from the application
* Display appropriate message when above functions fail
* Display appropriate message when Directory is empty
* Admin Access and each file’s desired location can be tracked and the date when it was created

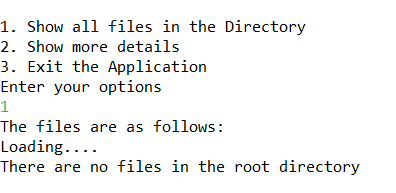
Product’s appearance: The product appearance includes following fields

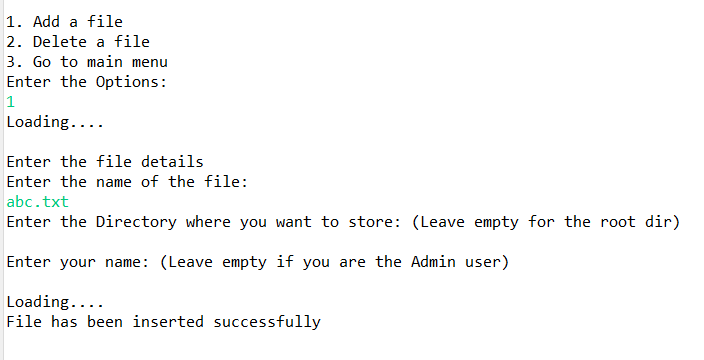
* Application name, the developer details, Greetings
* When launched the application ask admin name
* The details of the user interface such as options displaying the user interaction information as shown below
  1. The first option returns the current file names in ascending order
  2. The second option returns the details of the user interface such as
     + Add file
     + Delete file
     + Search file
     + Return to the main menu
  3. The third option to close the application
* Features to accept the user input to select one of the options listed

User interactions: As this is a prototyped application, the user interaction will be via a command line. The user must enter the correct option specified in the command line in order to perform various file operations.

Some of the screenshots are provided below:







# Number and duration of sprints required:

There will be three sprints each of one week.

Goal achieved in each sprint are provided below:

### In the first sprint we did the following

### 

* + Planning Modular approach to understand what concept can be modularized.
  + Understanding and building system architecture of the application

### In the Second sprint we improvised our function implementation for each functionality

* + Built basic UI for the application and to decide what are key parameters required for a basic file
  + Based on the parameter, the file class was developed

### In the Third sprint we updated or improvised design

* + UI improvement and code refactoring was decided upon this sprint
  + Built comparator class to compare each file attribut

# Flow of overall application (Life cycle of the project):

### Select and prioritize project

During the first step of the agile software development life cycle, the team scopes out and prioritizes the project. These priorities can be based on client’s requirement

**Diagram requirements for the initial sprint**

Once you have identified the project, work with stakeholders to determine requirements. You might want to use user flow diagrams or high-level UML diagrams to demonstrate how the new feature should function and how it will fit into your existing system.

**Construction/iteration**

Once a team has defined requirements for the initial sprint based on stakeholder feedback and requirements, the work begins

**Release the iteration into production**

You’re nearly ready to release your product into the world. Finish up this software iteration with the following steps:

* Test the Application
* Address any defects
* Finalize the user document
* Release the iteration through documentation

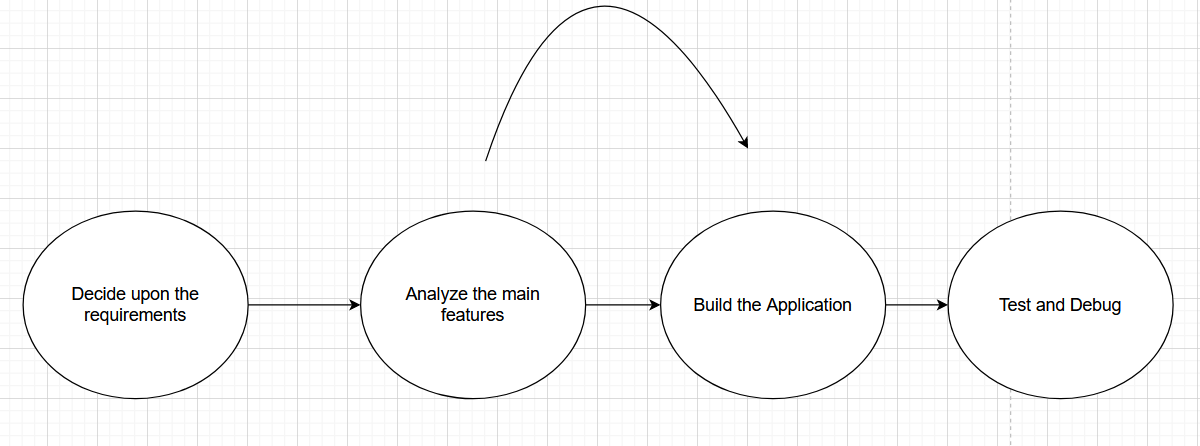
**Production and ongoing support for the software release**

This phase involves ongoing support for the software release. In other words, your team should keep the system running smoothly and show users how to use it. The production phase ends when support has ended or when the release is planned for retirement.

**Retirement**

During the retirement phase, you remove the system release from production, typically when you want to replace a system with a new release or when the system becomes redundant, obsolete, or contrary to your business model.

**Flow Of Application**



# Core Java Concepts Used:

* + **String Handling**: Strings, which are widely used in Java programming, are a sequence of characters. In Java programming language, strings are treated as objects. The Java platform provides the String class to create and manipulate strings.
  + **Usage of Threads**: To mimic asynchronous operation, use of Thread.sleep()
  + **Exception Handling:** The **Exception Handling in Java** is one of the

powerful mechanisms to handle the runtime errors so that the normal flow of the application can be maintained.

* + **Switch statements:** The Java switch statement executes one statement from multiple conditions. It is like [if-else-if](https://www.javatpoint.com/java-if-else) ladder statement. The switch statement works with byte, short, int, long, Enum types, String and some wrapper types like Byte, Short, Int, and Long. In other words, the switch statement tests the equality of a variable against multiple values.
  + **Modular Programming**: Each Class is separated in different folders, so debugging and other enhancement can be made easily
  + **Constructor overloading**: Based on the user’s interest different constructor gets called.
  + **Collection:** The **Collection in Java** is a framework that provides an

architecture to store and manipulate the group of objects. Collections can achieve all the operations that you perform on a data such as searching, sorting, insertion, manipulation, and deletion

# Data structure and algorithm used:

### ArrayList

`

The ArrayList class is a resizable array, which can be found in the java.util package.

The ArrayList in Java can have the duplicate elements also. It implements the List interface so we can use all the methods of the List interface here. The ArrayList maintains the insertion order internally.

It inherits the AbstractList class and implements List interface.

### Different ArrayList methods used:

* + **add():** This method appends the specified element to the end of this list
  + **indexOf():** Returns the index of particular instant of the object
  + **remove (int index):** This method removes the element at the specified position in this list. Shifts any subsequent elements to the left (subtracts one from their indices).

# Conclusion on enhancing the application:

# As this is a prototype of the application, it can be easily created using basic switch statement

# And basic function of ArrayList collection framework, where we add a file and delete a file based on the user input.

# Also, this application can be enhanced to work as per user requirement with some changes

# Project also includes various modules so that debugging is easy

# GitHub Repository Link:

https://github.com/kmadhav907/Java-Proj-1