**File Operation Application Project Report**

Project Details

Project Name: File Operation Project in Java

Developer: Rohit Singh

Date of Submission: 02nd September 2023

Table of Contents

[**File Operation Project Report** 1](#_Toc143983640)

[**Project Details** 1](#_Toc143983641)

[**1. Introduction** 1](#_Toc143983642)

[**2. Project Overview** 1](#_Toc143983643)

[**2.1 Project and Developer Details** 2](#_Toc143983644)

[**2.2 GitHub Repository** 2](#_Toc143983645)

[**3. Sprints and Achievements** 2](#_Toc143983646)

[**Sprint 1** 2](#_Toc143983647)

[**Sprint 2** 2](#_Toc143983648)

[**Sprint 3** 2](#_Toc143983649)

[**Sprint 4** 2](#_Toc143983650)

[**4. Algorithm and Flowcharts** 2](#_Toc143983651)

[**Algorithm** 3](#_Toc143983652)

[**Flowchart** 3](#_Toc143983653)

[**5. Core Concepts Used** 3](#_Toc143983654)

[**6. Project Enhancements** 3](#_Toc143983655)

[**7. Unique Selling Points** 3](#_Toc143983656)

[**8. Conclusion** 3](#_Toc143983657)

# 1. Introduction

## This report outlines the development process and key aspects of the LockedMe.com Prototype project. The objective of the project is to create a command-line prototype application that showcases the capabilities, appearance, and user interactions of the LockedMe.com platform. The application aims to implement fundamental file handling operations, such as viewing, adding, deleting, and searching files within a specified directory.

# 2. Project Overview

## As a Full Stack Developer, the primary goal is to complete the features of the application by planning the development in terms of sprints and then pushing the source code to a GitHub repository. The prototype focuses on command-line user interaction to simulate the LockedMe.com platform's functionality. By creating a user-friendly application, users can efficiently manage files. The project scope covers core file operations and sets the foundation for potential future enhancements.

## 2.1 Project Scope and Features

### The prototype's capabilities include viewing, adding, deleting, and searching files within a designated directory. The command-line interface ensures ease of use for users interacting with the application.

## 2.2 GitHub Repository

### The project's source code is available on the GitHub repository: <https://github.com/RohitSingh-NR/FileOperationsApplication/tree/main/src/com/fileOps> and please read the README.md file carefully to access the code properly before executing it.

# 3. Sprints and Achievements

## The project development was organized into three sprints, each focusing on specific features. The completed tasks are summarized below:

## Sprint 1

### Implemented File Viewing

### Implemented File Adding

## Sprint 2

### Implemented File Deletion

### Implemented File Searching

## Sprint 3

### Completed UI Refinement

### Conducted User Acceptance Testing

### Implemented Error Handling

### Conducted System Testing and Bug Fixes

# 4. Algorithm and Flowcharts

## Algorithm

## Initialize the application

## Ask user to provide root directory to use features of this application.

## Display the main menu to the user with options:

### View files in the directory

### Do some File Operation (Like Add, Delete and Search the file)

### Exit the application

## Wait for user input.

## If the user chooses to view files:

## List all files in the specified directory.

## Return to the main menu.

## If the user chooses to do some operation

## Display the sub-menu to the user with options:

## Add a new file

## Delete a file

## Search for a file

## Back to Main Menu

## Exit the application (Added this feature here also to find better usability)

## If the user chooses to add a new file:

## Prompt the user for the file name.

## Create a new file with the provided name in the specified directory.

## Return to the main menu.

## If the user chooses to delete a file:

## Prompt the user for the file name to be deleted.

## Check if the file exists; if yes, delete it.

## Return to the main menu.

## If the user chooses to search for a file:

## Prompt the user for the search criteria (e.g., file name with extension).

## Display the details of file matching the search criteria.

## Return to the main menu.

## If the user choose to return to main menu:

## Display the main menu option to user

## If the user chooses to exit:

## Terminate the application.

## If the user selects an invalid option, display an error message and return to the main menu.

## Flowchart

### Please refer attached document for Flowchart and open it as web layout in full screen to see it in correct way.

# 

User Defined path of Root Directory

Display Main menu

Take user input to choose one of the given options

# True

Show files from Specified Root Directory

Display files

# 

# False

Do some File Operations

# True True

Add files to the Specified Root Directory

Take user input to choose one of the given options

Add Files

# False

False True

Delete files from Specified Root Directory

Delete Files

Show file Details from Specified Root Directory

Exit the Application

Search files

True True

False

False

Back to main menu

True

Default

True

Invalid Input

False False

Default

Invalid Input

True

# 5. Core Concepts Used

## The project extensively employs Java's File and Directory classes for file operations. Modular programming is adopted to organize various functions. Additionally, the application integrates user interface design and error handling for a seamless user experience.

# 6. Project Enhancements

## The project is open to future enhancements for expanding its functionalities:

### Advanced Search Capabilities

### User Authentication and Security

### File Preview and Editing

### Multi-Directory Support

### Batch Operations

# 7. Unique Selling Points

## The File Operation Project offers several unique selling points that contribute to its significance and utility:

### User-Friendly Interface: The simplified design accommodates users of all skill levels.

### Hands-On Learning: It provides practical experience in Java file handling.

### Customizability and Extension: The modular structure allows for feature expansion.

### Real-World Relevance: The project simulates real file management scenarios.

### Foundation for Further Development: It serves as a starting point for more complex projects.

# 

# 8. Conclusion

## The LockedMe.com prototype effectively demonstrates file handling operations through a command-line interface. The planned sprints and the development process showcase the capabilities of a Full Stack Developer in creating practical applications. The development process enhanced Java programming skills and showcased the potential for building practical applications.