**Software Specification Document**

**LockedMe.com**

**22/07/2020**

**By: Sahana**

|  |
| --- |
| **Table of Contents** |

[1. Introduction 3](#_Toc244519333)

1.1 Purpose 3

1.3 Scope……………………………………………………………………………………….3

[2. General Description 3](#_Toc244519334)

2.1 Product Features 3

2.2 Design and Implementation 3

2.2.1 Algorithm…………………………………………………………………………....3

3. Conclusion……………………………………………………………………………………...5

1. Introduction

* 1. **Purpose:**

This is a software design specification for the LockedMe application. The Software Design Specification for LockedMe describes the system elements, functions and configurations necessary to properly operate the system with functional requirements outlined in the Functional Requirements Specification for the Locked me.

* 1. **Scope:**

LockedMe project to provide virtual key for repositories. The prototype of the application provides interaction through command line to work with the application.

## 2. General Description

**2.1 Product features:**

The application is designed using core Java which takes user input from command prompt. User can select any option from the list of operations of the application. User can select option to list the files in directory in ascending order, add a file to directory, delete a file from directory and search a file in directory. Depending on the input provided, respective operations will be performed and output is displayed on command prompt

**2.2 Design and implementation:**

**2.2.1 Algorithm:**

**procedure:** displayMainMenu()

**Input:** Option selected by user from menu displayed

**Output:** Perform operations based on user input

1. Display the options for file operations.

Display file names in ascending order.

Display file manipulation options

Exit from application

1. Read input

If input 🡪 1, go to displayFilesInAscending()

If input 🡪 2, go to displaySubMenu()

If input 🡪 3, exit from application

**procedure:** displayFilesInAscending()

**Output:** Files of current directory in alphabetical order

1. Get files from current directory
2. Go to sort files according to alphabetical order
3. List the files in alphabetical order

**procedure:** displaySubMenu()

**Input:** Option selected by user from submenu displayed

**Output:** Perform operations based on user input

1. Display the options for file manipulation.

Add a file to directory

Delete file from directory

Search for a file in directory

Navigate to main menu

1. Read input

If input 🡪 1, go to addFilesToDirectory()

If input 🡪 2, go to deleteFilesFromDirectory()

If input 🡪 3, go to searchFilesFromDirectory()

If input 🡪 4, navigate to main menu

**procedure:** addFilesToDirectory ()

**Input:** File name to be created

**Output:** Creating file in directory

1. Read file name to be created from the user
2. Get list of files from directory
3. Check if file exists in directory
4. If file exists, return ‘file exists’

Else

1. Create file in directory and return success message

**procedure:** deleteFilesFromDirectory ()

**Input:** File name to be deleted

**Output:** Deleting file from directory.

1. Read file name to be deleted from the user
2. Get list of files from directory
3. Check if file exists in directory
4. If file exists, delete the file from the directory and return ‘success’ message

Else

1. Return ‘unsuccessful’ message

**procedure:** searchFilesFromDirectory ()

**Input:** File name to be searched

**Output:** File search success/unsuccessful message.

1. Read file name to be searched from the user
2. Get list of files from directory
3. Check if file exists in directory
4. If file exists, return ‘success’ message

Else

5. Return ‘unsuccessful’ message

The application has four main functionalities as described above.

1. displayFilesInAscending: This method retrieves the files from current directory using file io operations and then sorts the files in alphabetical order. The method uses quick sort method to sort the files in alphabetical order.
2. addFilesToDirectory: This method takes filename from the user and adds the file to the current directory. Method first checks if the file exists in the directory. It uses exponential search technique to search for file. If the file doesn’t exist, then file is created using file io methods. If the file exists, then method return “file exists” message to command prompt.
3. deleteFilesFromDirectory: This method takes filename to be deleted from the user and deletes the file from the current directory. Method first checks if the file exists in the directory. It uses exponential search technique to search for file. If the file doesn’t exist, then “File Not Found” message displayed in the command prompt. It file exists, then file io method is used to delete the file.
4. searchFilesFromDirectory: This method takes filename to be searched from the user. It uses exponential search technique to search the file. If file exist then “success” response is displayed. If file does not exists then, “file not found” message is displayed.

## 3. Conclusion

The application is prototype of LockedMe.com which use command line for User interaction. The application can be further enhanced by providing a proper designed user interface which helps the user to interact with application in easier manner. Other functionalities like editing the file, renaming the file can also be included in the application in further enhancement.