**LOCKED ME**

**SPECIFICATION DOCUMENT**

**Introduction**

Company Lockers Pvt. Ltd aim to digitize their products and choose Lockedme.com as the first project to start with and a prototype of this application will be presented to the relevant stakeholders. Developed by Ramyasree.

**PLANNING DOCUMENT**

**PRODUCT BACKLOG**

**USER STORIES**

* 1. Display the welcome screen
  2. Retrieve the file names in an ascending order
  3. Perform Add/Delete/Search files and Navigate back to Main Menu
  4. Exit the Application

**SPRINT 1**

**SPRINT BACKLOG**

To add the following functionality to the application:

1. Display the welcome screen
2. Retrieve the file names in an ascending order
3. Exit the Application

**SPRINT PLANNING**

* 1. Display the welcome screen
     1. Application name and the developer details
     2. The details of the user interface such as options displaying the user interaction information
     3. Features to accept the user input to select one of the options listed
  2. Retrieve the file names in an ascending order
     1. The first option should return the current file names in ascending order.
  3. Exit the application
     1. There should be a third option to exit the application.

DEFINITION OF DONE

All the features in the sprint backlog would be considered done when all the tasks are completed

CAPACITY

Sprint capacity = 8hrs per day X 5 days X 1 = 40 hours

**SPRINT 2**

**SPRINT BACKLOG**

To add the following functionality to the application:

1. Perform Add/Delete/Search files and Navigate back to Main Menu

**SPRINT PLANNING MEETING**

USER STORIES TO TASKS

* 1. Add a user specified file to the application
     + 1. The second option should return the details of the user interface such as options displaying the following:
       2. Add a file to the existing directory list.
       3. You can ignore the case sensitivity of the file names.
  2. Delete a user specified file from the application
     + 1. The second option should return the details of the user interface such as options displaying the following:
       2. Delete a user specified file from the existing directory list.
       3. You can add the case sensitivity on the file name in order to ensure that the right file is deleted from the directory list.
       4. Return a message if FNF (File not found).
  3. Search a user specified file from the application
     + 1. The second option should return the details of the user interface such as options displaying the following:
       2. Search a user specified file from the main directory.
  4. Navigate back to Main Menu

DEFINITION OF DONE

All the features in the sprint backlog would be considered done when all the tasks are completed

CAPACITY

Sprint capacity = 8hrs per day X 5 days X 1 = 40 hours

**Algorithm:**

Step1: Display Welcome Screen and Read User Input

Step 2: If Input is 1, then retrieve files from the current directory

Step 3: If Input is 2, then give the user options to add, delete, search or return to the main menu.

Step 4: If Input is 3, Exit the application.

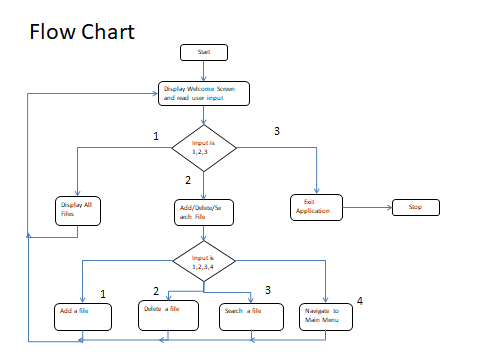
Step 5: When the user Input is 2 in Step 2, and then choice is 1, add a user specified file

Step 6: When the user Input is 2 in Step 2, and then choice is 2, delete a user specified file

Step 7: When the user Input is 2 in Step 2, and then choice is 3, search a user specified file

Step 8: When the user Input is 2 in Step 2, and then choice is 4, go back to the main menu.

**Flowchart:**

****

**CORE CONCEPTS USED IN THIS PROJECT:**

Main.java

This the main class which is the entry point of the application.

InputValidations.java

This is the abstract layer interface for validations of user input

InputValidationsImpl.java

This is the implementation class for the validations of user input

Message.java

This class handles all the printing to the console.

FileOperations.java

This is the data access object interface which has all the methods for file operations like add, delete, and search.

FileOperationsImpl.java

This is the implementation of file operations interface.

LockedMeException.java

This is used for custom Exception

**Github Repositry Link:**

<https://github.com/KarnamR/PhaseOne>

**Conclusion:**

This basic prototype can be enhanced by adding a UI for the user to choose the Location and Provide only limited access for the folders and files (Read Only/ Edit / Delete) based on the role of users.