**LOCKEDME PROJECT USING ECLIPSE**

**OBJECTIVE OF THE PROJECT**:

To develop a prototype of the application and digitize their products as a full stack developer for Company Lockers Pvt Ltd.

1. Specification document - Product’s capabilities, appearance, and user interactions
2. Number and duration of sprints required
3. Setting up Git and GitHub account to store and track your enhancements of the prototype
4. Java concepts being used in the project
5. Data Structures where sorting and searching techniques are used.
6. 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

**FLOW AND FEATURES OF THE APPLICATION:**

**I had planned for 3 sprints**

* Display Menu
* Cases for creating, searching deleting & displaying the files
* Code for developing the file creation, searching, deleting and displaying

**Flow of the application**

* Displaying the Menu with all the options available for the user or customer to choose the respective operation performed.
* Based on the choice or input from the user the respective case will be activated.
* The options are as below

1. Display all the files
2. Create a file
3. Delete a file
4. Search for a file
5. Exit

* If the user selects the option “1” then the output will be displaying the files present in the Projects path.
* If the user selects the option “2” then the output will be asking to generate the required file.
* If the user selects the option “3” then the output will be asked to delete the file requested.
* If the user selects the option “4” then the output will be given whether the input file is present the project path or not.
* The last option “5” will be used to exit from the loop.
* If the user gives other input, then the output be given as Invalid answer.
* Let us see the same in a flow chart for a detailed explanation

Choose options Other than (1-5)

1 2 3 4 5

Display all files

Search a File

Create a File

Delete a File

Exit

Invalid

L=1

IF IF FOR

Enter File Name

F

files

T F T F IF

No files Avl

Print all Files

Txt

Lines

Created succ

File

Del

No File

file

T F

File Avl

File Not Avl

Exception

ErroMessage

CLOSE

**LIST OF CORE CONCEPTS AND ALGORITMS USED:**

**static final** :- This keywords are used whenever the developer wants a field to be fixed. In this currurent project we would like fix the fields below

**static** **final** String ***projectFilesPath***

To fix the path of the files in a desired location

**static** **final** String ***errorMessage***

To get a fixed output whenever there is a exception in the program or the application

**Scanner**

The Scanner class is used to get user input, and it is found in the java.util package. To use the Scanner class, create an object of the class and use any of the available methods found in the Scanner class documentation. In our example, we will use the nextLine() method, which is used to read Strings.

**Switch and case statement**

The switch statement is used to switch from different cases. Here we have used 4 cases

* + - 1. To display all the files
      2. To create a file
      3. To delete a file and
      4. To search a file
      5. Exit

**GetAllfiles**

This directory is to get all the files in the file location

**public** **static** **void** getAllFiles()

**ListOfFlies**

listOfFiles.length==0

This directory is used to get the list in a ascending sorting way.

**LinesCount**

This is used to count no of lines in the file present in a file

.**write**

This is used to write something in an existing/created file.

**.delete**

This is used to delete a selected file in a location

.**contains**

This is used to search or find a particular file in a respective path

**Try and catch**

These are used when an exception is handled and are catched to give a user defined output

**Finally**

This is used to give an output of final after all the exceptions are handled.

In the current program we are using finally to close the program.

**close**

This command is used to close the loop for which the program is running continuously.

**CONCLUSION on ENHANCING:**

* Organisations can be use this application for example
* HR department can use the application when ever a new has been added to organisation. They can add, search and delete his profile if he leaves.
* Project management department can use this application to add new tasks and delete as soon as the task is completed or can maintain a data for future reference to retrieve/search the data.
* These are some befits and unique selling points.
* Application can be more enhanced one the user uses and give the feedback for more specific features.