# Project and Developer Details:

* Project – Prototype of Application to be build in terms of sprints later push to git Hub repository.
* Developer – Surya Senthil

# Sprints planned and the tasks achieved:

The project is to be completed in two sprints.

* Tasks:
  + Analysing the concept of the project.
  + Planning the flow of the application.
  + Setting up Git and GitHub to track the prototype development.
  + Core Concepts of java program used to develop the project.
  + Testing the program with different user inputs.

# Algorithms and flowcharts of the application:

Choose the path in Local Pc

Print Welcome Screen and Developer details

Display Main Menu and Take user input

Enter into switch case main menu

Case 1

Retrieving the file names in an ascending order in path given

True

False

Case 2

Performing the Secondary Sub menu and take user input

True

False

Case 3

True

Program exited Successfully

Exit

Default: Display message to give correct input

Sub menu will be displayed to perform file operations

Switch case

User can add file inside the folder by entering file name.

Case 1

True

False

User can Delete the file by giving the input of the file name.

Case 2

True

False

User can search for the file name inside the folder.

Case 3

True

False

Return to Main Menu

Case 4

True

Default: Display message to give correct input

# Core concepts used in project:

* Abstraction, Flow control, Exception handling, Switch case, File handling.

# Product’s capabilities, appearance, and user interactions

## Step 1: Creating a new project

* Create a new java project in Eclipse
* Project name as Lockers.Pvt. Ltd
* Create a new package with name as a com.LockedMe.com.
* Under the package create a new class LockedMain.java

## Step 2: Writing program in LockedMian

package com.LockedMe.com;

import java.io.IOException;

/\*Abstract class contains the menu of the Protoype project LockedMe\*/

public abstract class LockedMain {

public static void main(String[] args) throws IOException { //Main class

WelcomeScreen.*Info*(); //Class created with method name

FileOperations.*info*();

}

}

## Step 3: Writing a program in java to display WelcomeScreen and Menu options

* Create a new class with name WelcomeScreen.java
* Writing a new method info under WelcomeScreen.java

package com.LockedMe.com;

public class WelcomeScreen {

public static void Info() {

//Welcome screen with info of the Application name and developer name

System.***out***.println("^^^^^^^^^^^^Lockedme.com^^^^^^^^^^^^^^\n" +

"This application is developed by surya\n"

+ "---------------------------------------");

String Welcomeboard = "Features and three operations of the application\n\n" +

"1.Retrieving the file names in an ascending order.\n"+

"2.Option to add,delete,search a user specified file from the application.\n"+

"3.Closing the application.";

System.***out***.println(Welcomeboard);

}

public static void mainmenu () {

//main menu options to choose

System.***out***.println("\n-------------------------------");

String menu = "Select any option number from below\n"

+ "1. Retrive all files\n"

+ "2. Display menu for file operations\n"

+ "3. Exit program";

System.***out***.println(menu);

}

Output:

^^^^^^^^^^^^Lockedme.com^^^^^^^^^^^^^^

This application is developed by surya

---------------------------------------

Features and three operations of the application

1.Retrieving the file names in an ascending order.

2.Option to add,delete,search a user specified file from the application.

3.Closing the application.

-------------------------------

Select any option number from below

1.Retrive all files

2.Display menu for file operations

3.Exit program

## Step 4: Writing a program in java to display SubMenu options

public static void submenu() {

//sub menu options to choose

String nextmenu = "Select any option number from below\n"

+ "1. Add a file\n"

+ "2. Delete a file\n"

+ "3. Search a file\n"

+ "4. Exit submenu";

System.***out***.println(nextmenu);

}

}

## Step 5: Writing a program in java to perform file operations

* Create a new class with name FileOperations.java
* Writing a new method info under FileOperations.java
* Writing a switch case to perform the first operation

public class FileOperations {

public static void info() throws IOException {

Scanner scan = new Scanner(System.***in***);

while(true) {

WelcomeScreen.*mainmenu*();

int ch;

ch = scan.nextInt();

switch(ch) {

case 1:

String path = "F:\\Simpli\_Project\\";

File file = new File(path);

File[] filename = file.listFiles();

for(File ff:filename) {

System.***out***.println(ff.getName());

}

System.***out***.println("From folder - " + path );

break;

* Writing a nested switch case in case 2

case 2:

WelcomeScreen.*submenu*();

int ss;

ss = scan. nextInt ();

switch(ss) {

case 1:

//Adding new file

String Addpath = "F:\\Simpli\_Project\\";

System.***out***.println("Enter filename to create: ");

String CreateFile = scan. next ();

String Finalpath = Addpath+CreateFile;

File AddFile = new File (Finalpath);

if (AddFile.createNewFile()) {

System.***out***.println("File created at " + Addpath + CreateFile);

}

else {

System.***out***.println("File not created");

}

break;

case 2:

//Delete an existing file

String Delepath = "F:\\Simpli\_Project\\";

System.***out***.println("Enter file name to delete: ");

String DeleteFile = scan. next ();

String delepath = Delepath+DeleteFile;

File Delefile = new File(delepath);

System.***out***.println(Delefile);

if (Delefile.delete()) {

System.***out***.println(DeleteFile + " File deleted");

}

else {

System.***out***.println("file not deleted");

}

break;

case 3:

//search a file inside a folder

String Searchpath = "F:\\Simpli\_Project\\";

System.***out***.println("Enter file name to search: ");

String sear = scan. next ();

File Filesearch = new File (Searchpath);

File [] itreate = Filesearch.listFiles();

int flag = 0;

for (File ff: itreate) {

if (ff. getName (). equalsIgnoreCase(sear)) {

flag = 1;

break;

}

else {

flag = 0;

}

}

if (flag ==1) {

System.***out***.println("File is found " + Searchpath + sear);

}

else {

System.***out***.println("file not found");

}

break;

case 4:

//Return to main menu

System.***out***.println("program exited Submenu");

break;

default:

System.***out***.println("please select a valid number");

}

break;

* Case 1: used to perform of creating a new file inside the folder when the user gives a file name
* Case 2: used to delete a file when the user inputs the file name
* Case 3: it is used to perform when a user inputs the name to search the folder to show the file
* Case 4: return to the main menu
* Default: when invalid number number is given default is invoked.

case 3:

//Exiting the program

System.***out***.println("Program exited succesfully");

System.*exit*(0);

default:

System.***out***.println("please select a valid number");

}

}

}

}

Case 3: case 3 in main menu is to perform of the exit o fthe program

Output:

* Retrive all files by giving user input 1

^^^^^^^^^^^^Lockedme.com^^^^^^^^^^^^^^

This application is developed by surya

---------------------------------------

Features and three operations of the application

1.Retrieving the file names in an ascending order.

2.Option to add, delete, search a user specified file from the application.

3.Closing the application.

-------------------------------

Select any option number from below

1.Retrive all files

2.Display menu for file operations

3.Exit program

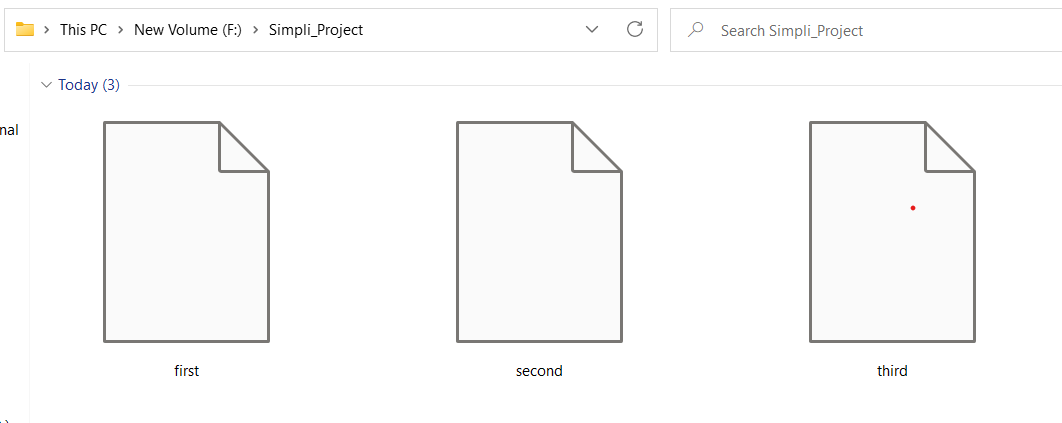
1

first

second

Third

From folder - F:\Simpli\_Project\



* User input 2 to display the sub menu to add, delete, search, return to main menu options

-------------------------------

Select any option number from below

1.Retrive all files

2.Display menu for file operations

3.Exit program

2

Select any option number from below

1.Add a file

2.Delete a file

3.Search a file

4.Exit submenu

* By giving user input 1 in the sub menu, we can create a file by giving a file name

Select any option number from below

1.Add a file

2.Delete a file

3.Search a file

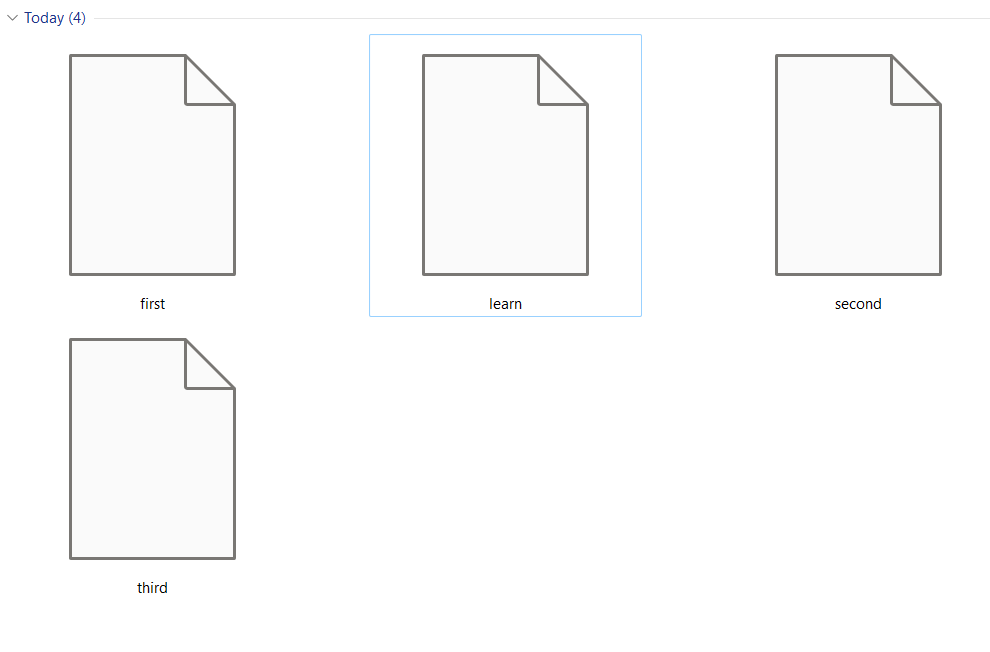
4.Exit submenu

1

Enter filename to create:

Learn

File created at F:\Simpli\_Project\Learn



* User input 2 in sub menu can delete the file by delete file operation

Select any option number from below

1.Add a file

2.Delete a file

3.Search a file

4.Exit submenu

2

Enter file name to delete:

learn

F:\Simpli\_Project\learn

learn File deleted

* User input 3 to search for a file in folder

Select any option number from below

1.Add a file

2.Delete a file

3.Search a file

4.Exit submenu

3

Enter file name to search:

first

File is found F:\Simpli\_Project\first

* User input 4 to return to main menu

Select any option number from below

1.Retrive all files

2.Display menu for file operations

3.Exit program

2

Select any option number from below

1.Add a file

2.Delete a file

3.Search a file

4.Exit submenu

4

program exited Submenu

-------------------------------

Select any option number from below

1.Retrive all files

2.Display menu for file operations

3.Exit program

* Case 3 of main menu is to perform Exit the program

-------------------------------

Select any option number from below

1.Retrive all files

2.Display menu for file operations

3.Exit program

3

Program exited succesfully