

## Project 1 - LockedMe – Virtual Key for Repositories.

### Project and developer details

The Project is developed by Rishabh Dixit

The all code of project is hosted on GitHub:- <https://github.com/dixitrishabh/LockedMe-Projectv1>

### Sprints planned and the tasks achieved in them

Creating a document and file structure of the project.

Firstly, I Initialized a git reop.

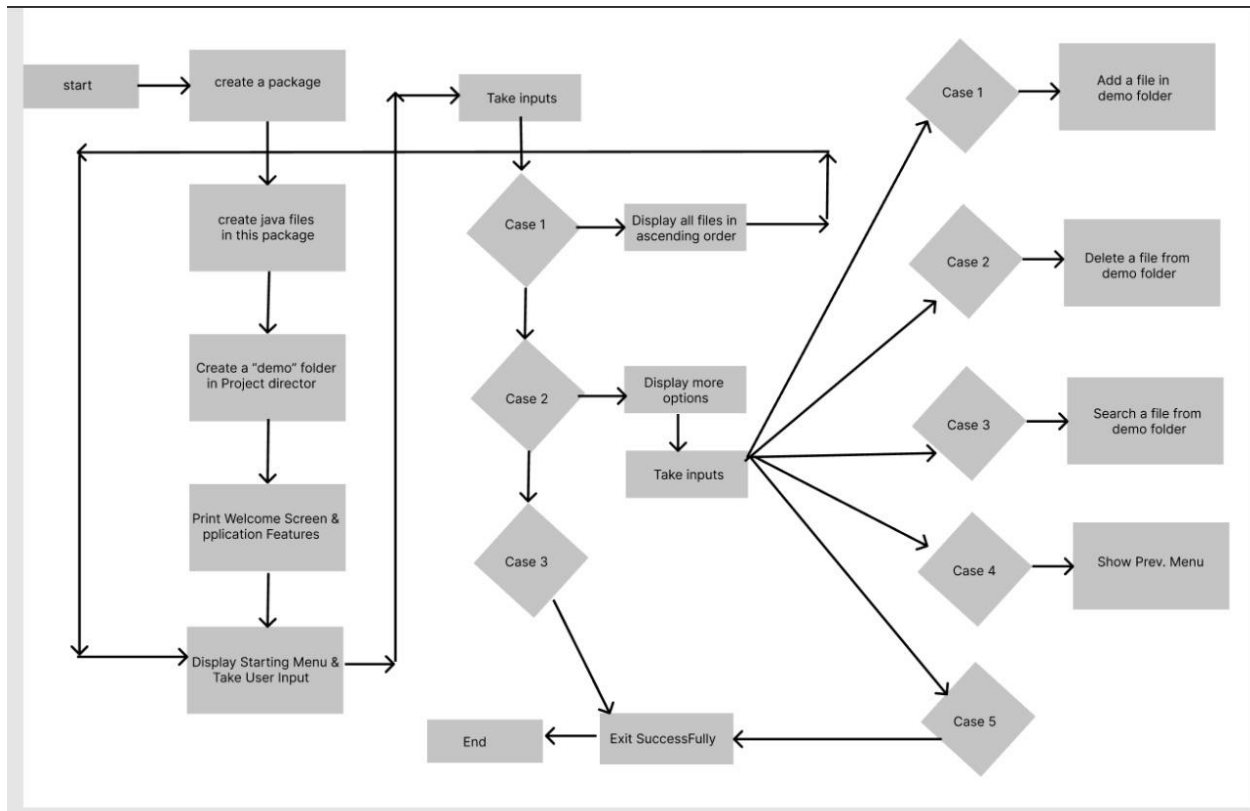
Writing a java programs to complete this java application.

After completing all code, I have performed different test case on this java application.

At last, I push all correct to GitHub (also pushing in between of the project).

### Algorithms and flowcharts of the application Core concepts used in the project

Recursion, Exception Handling, Searching, Sorting, File Handling, Collections Frameworks.

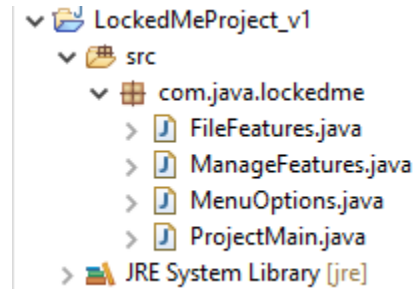


## Project Details: -

Create a java project in Eclipse

Create a package com.java.lockedme

Create files names



**Write a java code or create a main method in ProjectMain to run the application.**

```
1 package com.java.lockedme;
2
3 public class ProjectMain {
4
5     public static void main(String[] args) {
6
7         //Creating a demo if its is not there.
8         FileFeatures.makeDemoFolder("demo");
9
10        //Application Info.
11        MenuOptions.printWelcomeScreen("LockedMe", "Rishabh Dixit");
12
13        ManageFeatures.manageWelcomeMessage();
14    }
15 }
16
17 }
```

## Write a java code in MenuOption to give User to perform actions.

```
package com.java.lockedme;

public class MenuOptions {

    public static void printWelcomeScreen(String applicationName, String developerName) {
        String applicationDeatils = String.format("-- Welcome to %s.com-- \n" + "--Application is developed by %s--.\n"
            + "-----\n", applicationName, developerName);

        String appliactionFunction = "You can use this appliaction to :-\n"
            + "* Retrive all file names in the \"demo\" folder\n"
            + "* ADD Search, Delete files in \"demo\" folder\n"
            + "* Please provide the correct file name for delete and search\n";

        System.out.println(applicationDeatils);

        System.out.println(appliactionFunction);
    }

    public static void displayMenu() {
        String menu = "\n-----Select option from below and press Enter-----\n"
            + "1. Retrieve all files \"demo\" folder \n"
            + "2. Display menu for file features \n"
            + "3. Exit Program\n";
        System.out.println(menu);
    }

    public static void displayMenuOptions() {
        String fileMenuOptions = "\n Select any option from below and presss Enter\n"
            + "1. Add a file to \"demo\" folder\n"
            + "2. Delete a file from \"demo\" folder\n"
            + "3. Search a file \"demo\" folder \n"
            + "4. Show Prev. Menu\n"
            + "5. Exit Program\n";

        System.out.println(fileMenuOptions);
    }
}
```

Write a java code in ManageFeatures to handle all the option in this application.

```
do {
    try {
        MenuOptions.displayMenu();
        int input = sc.nextInt();

        switch(input) {
            case 1:
                FileFeatures.displayAllFiles("demo");
                break;
            case 2:
                ManageFeatures.manageMenuOptions();
                break;
            case 3:
                System.out.println("Program exit successsully");
                running = false;
                sc.close();
                System.exit(0);
                break;
            default:
                System.out.println("Please select the options that provided");
        }
    } catch (Exception e) {
        System.out.println(e.getClass().getName());
        manageWelcomeMessage();
    }
} while(running == true);
}
```

```

switch(input) {
case 1:
    System.out.println("Enter the name of file which you have to add \"demo\" folder");
    String AddFile = sc.next();

    FileFeatures.creatingFile(AddFile, sc);

    break;
case 2:
    System.out.print("Enter the name of file that you have to delete \"demo\" folder");
    String deleteFile = sc.next();

    FileFeatures.makeDemoFolder("demo");
    List<String> deleteFiles = FileFeatures.displayLocationofFiles(deleteFile, "demo");

    String deletePrompt = " \n(Enter 0 if you want to delete)";

    System.out.println(deletePrompt);
    |

    int idx = sc.nextInt();

    if (idx != 0) {
        FileFeatures.deleteaFile(deleteFiles.get(idx - 1));
    }else {
        for(String path : deleteFiles) {
            FileFeatures.deleteaFile(path);
        }
    }

    break;

case 3:
    System.out.println("Enter the File that you have to search \"main\" folder ");
    String fileName = sc.next();

    FileFeatures.makeDemoFolder("demo");

case 3:
    System.out.println("Enter the File that you have to search \"main\" folder ");
    String fileName = sc.next();

    FileFeatures.makeDemoFolder("demo");
    FileFeatures.displayLocationofFiles(fileName, "demo");

    break;

case 4:
    return ;

case 5:

    System.out.println("Program exit successfully");
    running = false;
    sc.close();
    System.exit(0);
default:
    System.out.println("Please select the option that is provided");

}

}catch(Exception e) {
    System.out.println(e.getClass().getName());
    manageMenuOptions();
}
}

while(running == true);

```

## Write a java code in FileFeatures to perform all operation in the application.

```
//making a demo folder
public static void makeDemoFolder(String folderName) {
    File file = new File(folderName);

    if(!file.exists()) {
        file.mkdirs();
    }
}

//display all files
public static void displayAllFiles(String path) {
    FileFeatures.makeDemoFolder("demo");

    List<String> filesNames = FileFeatures.listFilesInDirectory(path, 0, new ArrayList<String>());

    System.out.println("Display all Files in ascending order");
    Collections.sort(filesNames);

    filesNames.stream().forEach(System.out::println);
}

//list all files
public static List<String> listFilesInDirectory(String path, int indentationCount, List<String> filesNames){
    File dir = new File(path);
    File[] files = dir.listFiles();
    List<File> fileList = Arrays.asList(files);

    Collections.sort(fileList);

    //Adding a new file in demo folder
    public static void creatingFile(String AddFile, Scanner sc) {
        FileFeatures.makeDemoFolder("demo");
        Path Filepath = Paths.get("./demo/" + AddFile);

        try {
            Files.createDirectories(Filepath.getParent());
            Files.createFile(Filepath);
            System.out.println(AddFile + " created successfully ");

            System.out.println("Would you want add content in the file ? (Y/N)");
            String choice = sc.next().toLowerCase();

            sc.nextLine();

            if(choice.equals("y")) {
                System.out.println("\nInput content and Press enter\n");
                String content = sc.nextLine();
                Files.write(Filepath, content.getBytes());
                System.out.println("Written content in file " + AddFile);
                System.out.println("Content can be read by Notepad");
            }
        } catch (IOException e) {
            System.out.println("File created failed" + AddFile);
            System.out.println(e.getClass().getName());
        }
    }

    //Display the location of the file.
    public static List<String> displayLocationofFiles(String fileName, String path){
        List<String> fileNameList = new ArrayList<>();
        FileFeatures.searchFile(path, fileName, fileNameList);
    }
}
```

```

//Searching a file by its name
public static void searchaFile(String path,String fileName, List<String> fileName) {
    File dir = new File(path);
    File[] files = dir.listFiles();
    List<File> fileList = Arrays.asList(files);

    if(files != null && files.length > 0) {
        for(File file : fileList) {
            if(file.getName().startsWith(fileName)) {
                fileName.add(file.getAbsolutePath());
            }

            if(file.isDirectory()) {
                searchaFile(file.getAbsolutePath(),fileName,fileName);
            }
        }
    }
}

//Deleting a file
public static void deleteaFile(String path) {

    File currentFile = new File(path);
    File[] files = currentFile.listFiles();

    if(files != null && files.length > 0) {
        for (File file : files) {

            String fileName = file.getName() + " at " + file.getParent();
            if(file.isDirectory()) {
                deleteaFile(file.getAbsolutePath());
            }
        }
    }
}

```

**GitHub Repo link :-** <https://github.com/dixitrishabh/LockedMe-Projectv1>

### **Your conclusion on enhancing the application and defining the USPs (Unique Selling Points):**

Allowing user to add a file in demo folder.

Allowing user to user to delete and search a given file from its name.

Allowing user to retrieve all file from demo folder.

The files will be sort in ascending order.

### **Some USPs –**

The application works very smoothly while taking inputs from the user.

The user can add, delete and search a file.

The user moves back in the main menu.

All files will be sort in Ascending order.

The application handle mismatch input exceptions.