**LockedMe Project – Virtual Key for Repositories**

This document contains sections for:

* [Sprint planning and Task completion](#Sprint_plan)
* [Core concepts used in project](#Core_concepts)
* [Flow of the Application](#Flow).
* [Demonstrating the product capabilities, appearance, and user interactions.](#Product_capability)
* [Unique Selling Points of the Application](#USP)
* [Conclusions](#Conclusions)

The code for this project is hosted at <https://github.com/sitansusubudhi/LockedMe>.

The project is developed by Sitansu Subudhi.

## Sprints planning and Task completion

The project is planned to be completed in 1 sprint (Calculated each sprint is 7 days). Tasks assumed to be completed in the sprint are:

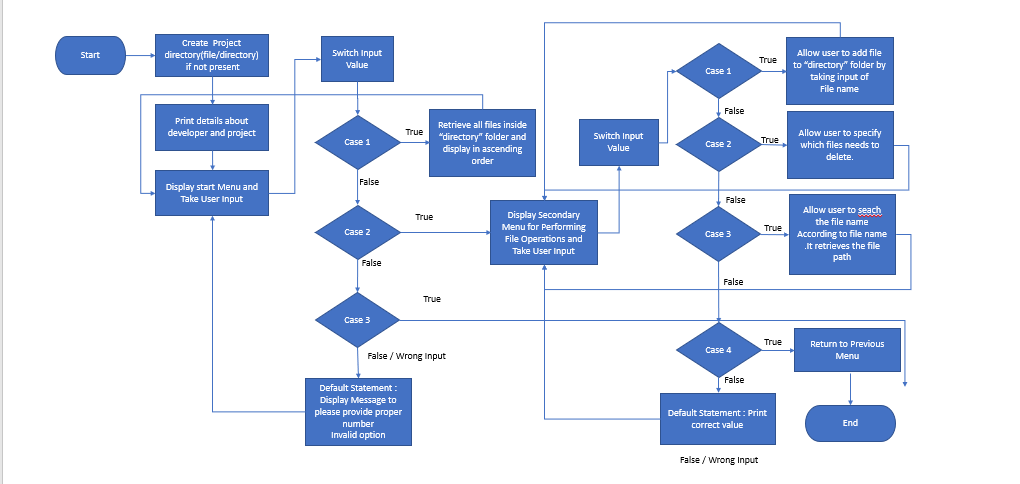
* Creating the flow of the application.
* Creating git repository virtualKey to track changes as development progresses.
* R&D for to understand the project and implemented.
* Testing the project with user input manually.
* Pushing code to GitHub Repository.
* Creating this specification document highlighting application capabilities, appearance, and user interactions.

## Core concepts used in project

* File Handling
* Collection Framework & Sorting Algorithm.
* Basic java feature (switch -if else, util package etc.)
* OOPS concepts.
* Exception Handling.
* Flow Diagram.

Collections framework, File Handling, Sorting, Flow Control, Recursion, Exception Handling, Streams API

## Flow of the Application

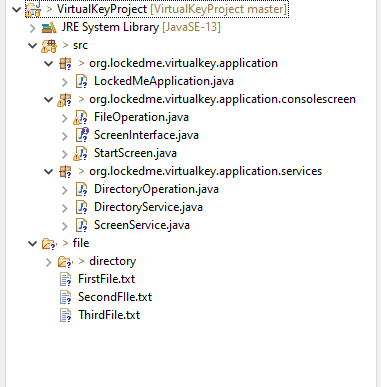


## Demonstrating the product functionalities, appearance, and user interactions

## **Step 1:** Creating a new project in Eclipse

* Open Eclipse
* Go to File -> New -> Project -> Java Project -> Next.
* Type in any project name and click on “Finish.”
* Select your project and go to File -> New -> Class.
* Creates the packages for Application (Entry-point of program), Console-based interaction, and Services.
* Implement the project according to packages.

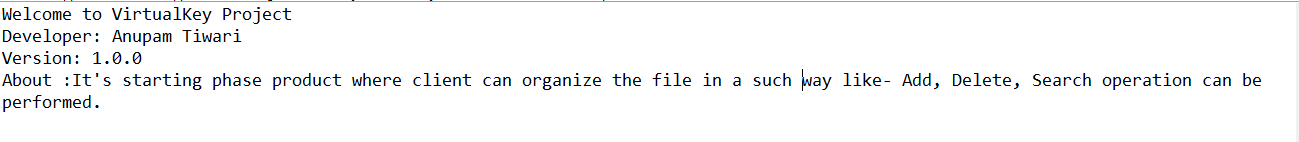
## **Step 2:** File Structure



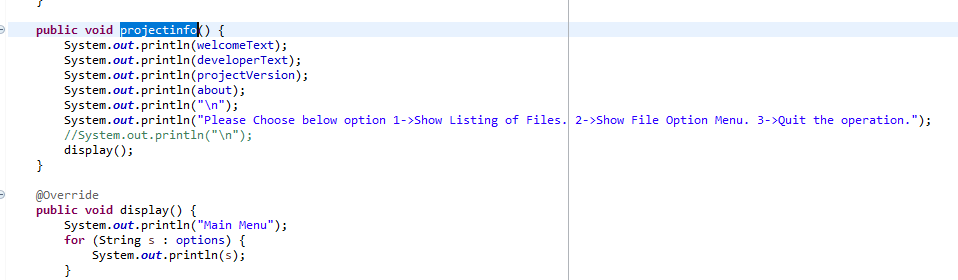
## **Step 3:** Project Demonstration

3.1 Application name and the developer details.

output

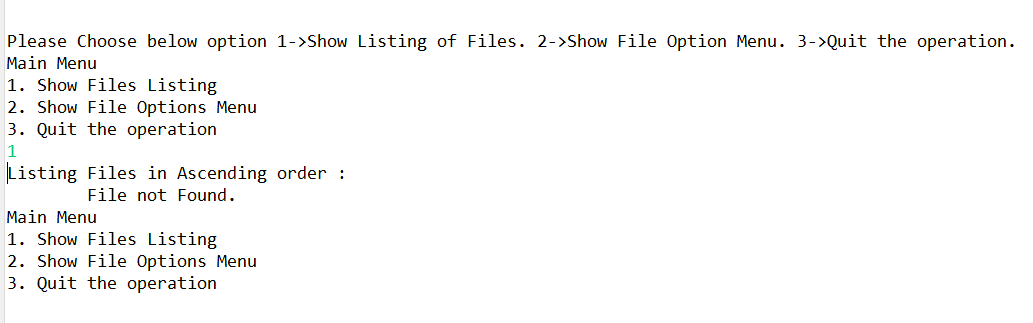


Implementation logic-

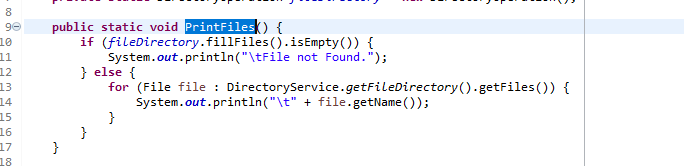


3.2 If Folder is Empty output will be file not found.

Output

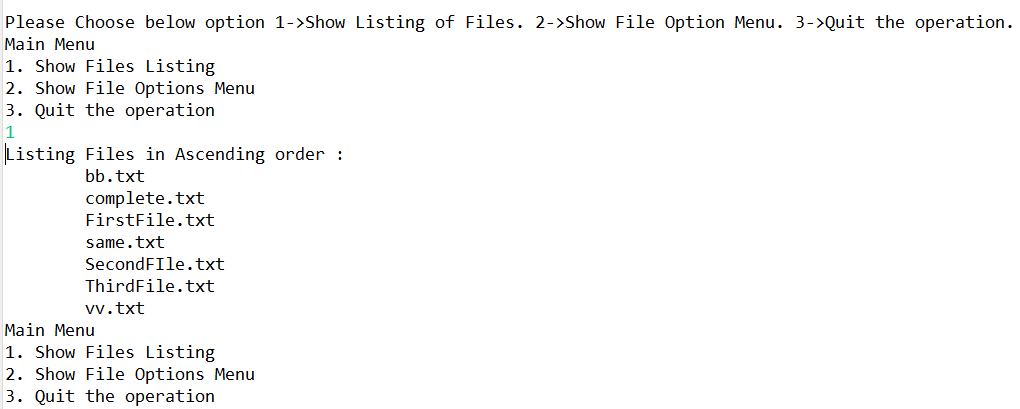


Implementation logic-

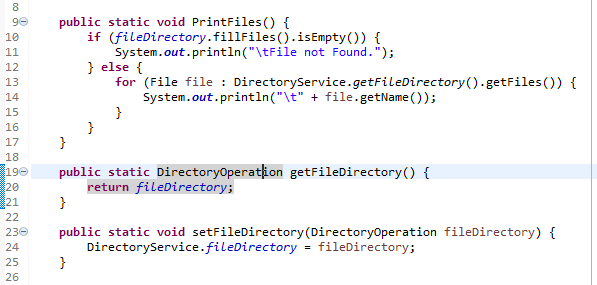


3.3 User can retrieve all files in Ascending order.

Output

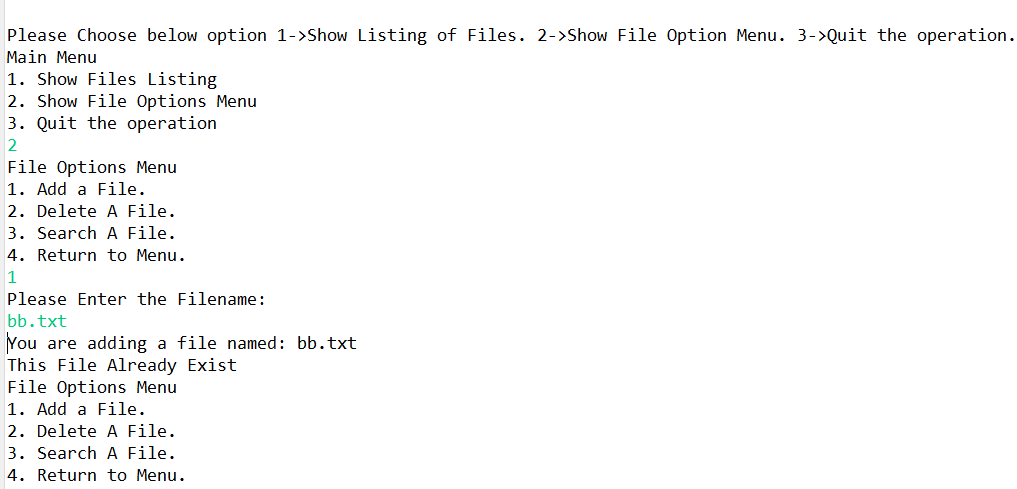


Implementation logic-

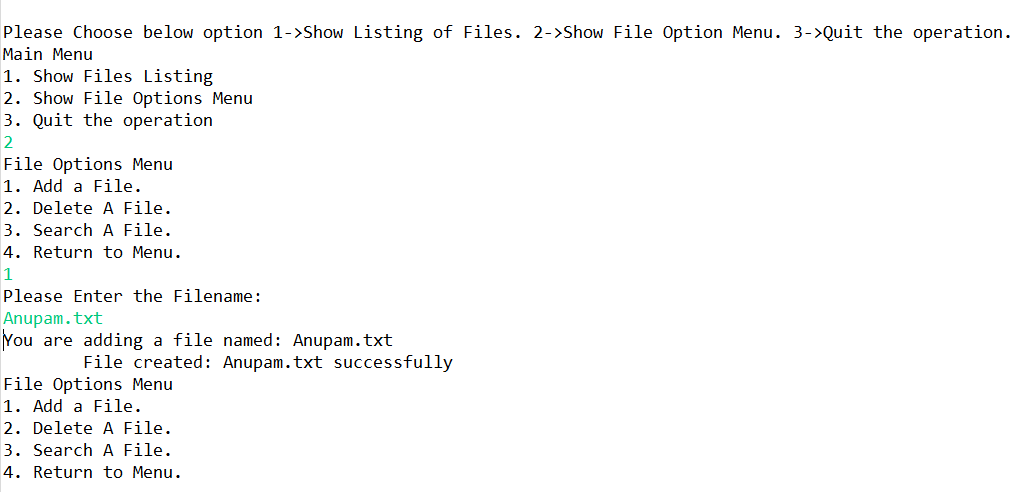


3.4 User can Add files.

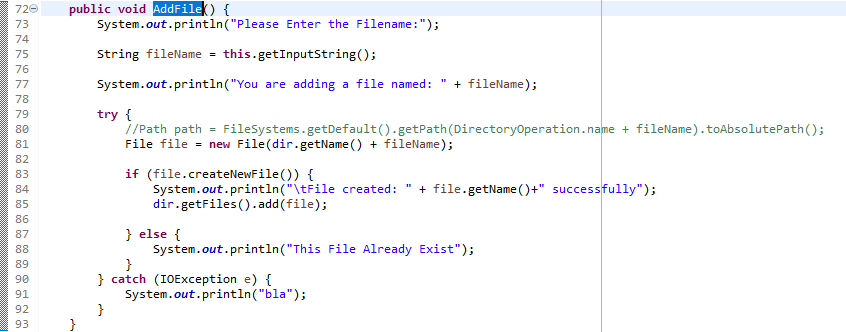
Output1 (If File already exist It will not create any new file.)



Output2 (create new file. Add Successfully)

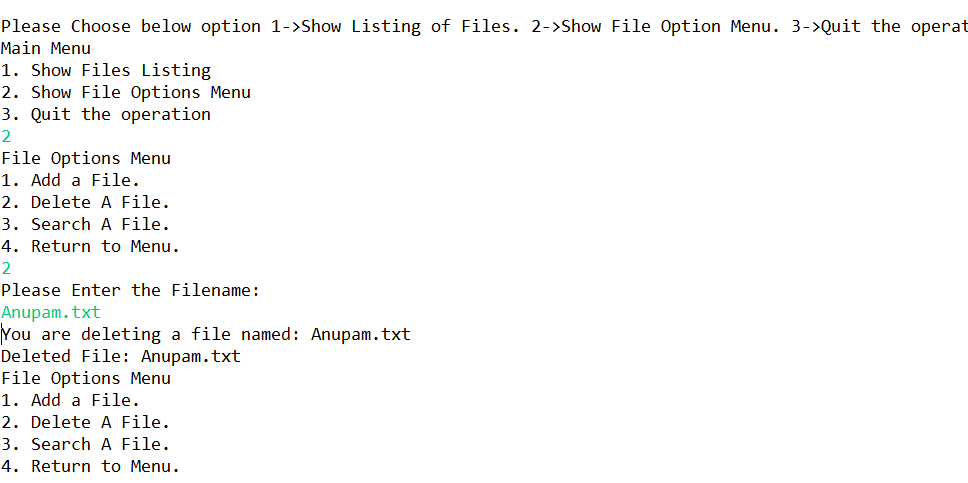


Implementation logic-

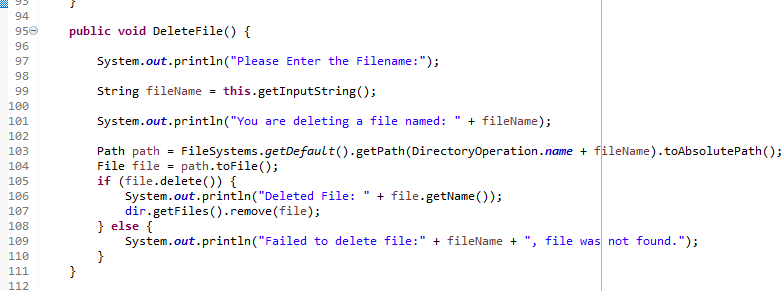


3.4 User can Delete specific files.

Output (File deleted successfully.)

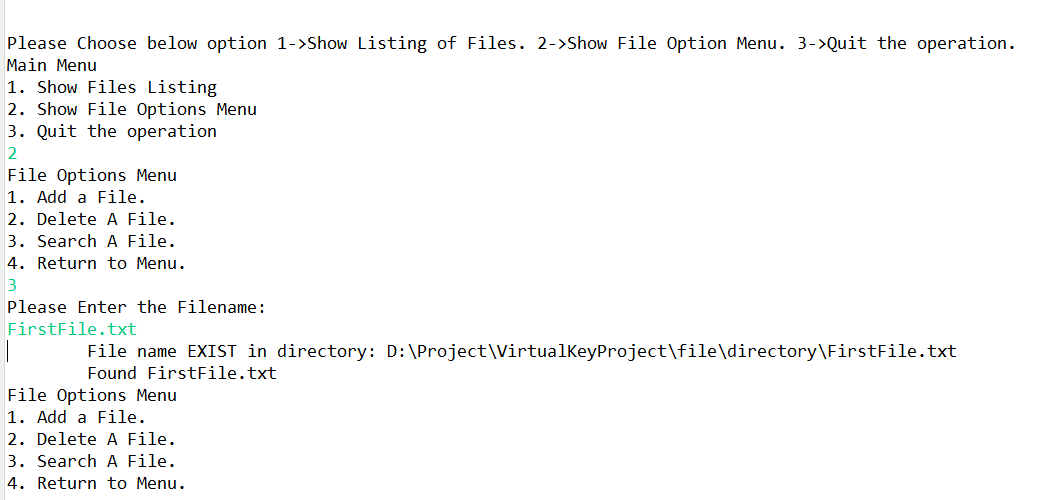


Implementation logic-

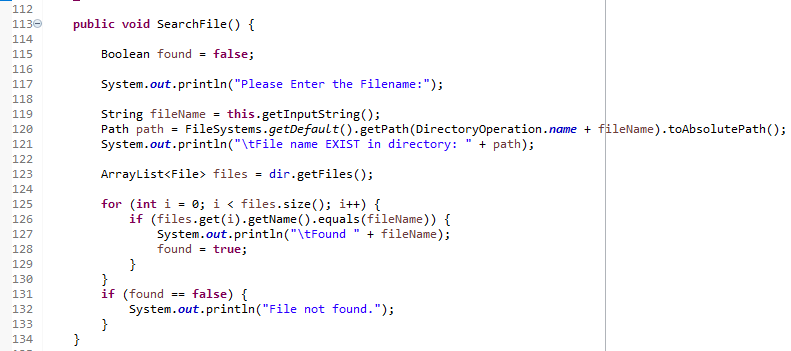


3.4 User can Search specific files.

Output (File search for path and name successfully.)

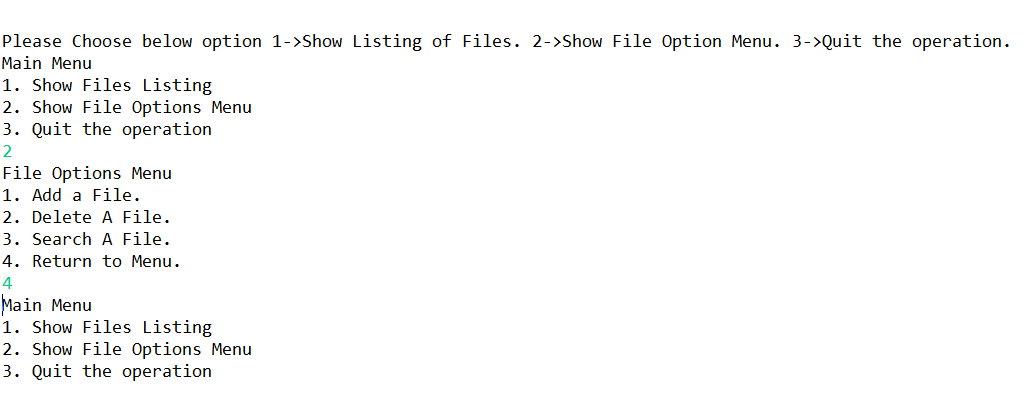


Implementation logic-

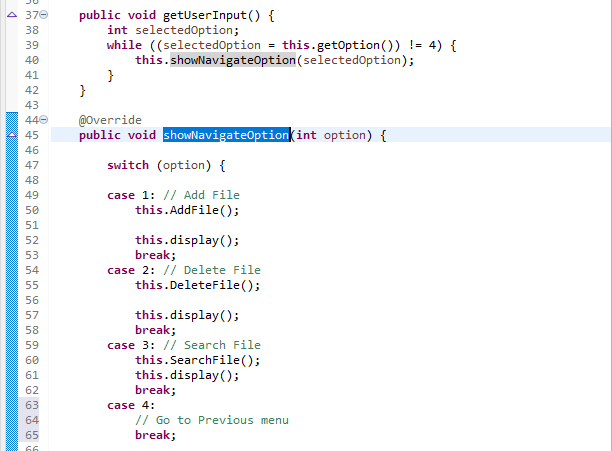


3.5 User After completion of all operations (Add, Search, Delete) can return to the main menu.

Output

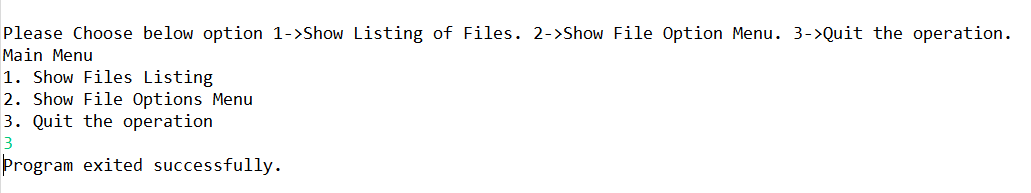


Implementation logic-



3.6 User can quit the operation.

Output



Implementation logic-



## **Step 4:** Pushing the code to GitHub repository

* Open your command prompt and navigate to the folder where you have created your files.

**cd <folder path> (Eclipse workspace)**

* Initialize repository using the following command:

**git init (Create repository)**

* Add all the files to your git repository using the following command:

**git add . (All Files)**

**git add -p (paragraph)**

* Commit the changes using the following command:

**git commit . -m <commit message>**

* Push the files to the folder you initially created using the following command:

**Git pull origin master (up-to-date)**

**git push -u origin master**

## Unique Selling Points of the Application

1. Organize all Files in one place. All operation can control in one place. No needs to do it manually. It saves time.
2. This functionality can we use any enterprise project where file handling is used a lot like application related Education sector. Where we can perform operation on the data.
3. User is also provided the option to search content. Not only Filename but user can see the path of file.
4. The application also allows user to delete folders which are not empty.
5. The user is able to seamlessly switch between options or return to previous menu even after any required operation like adding, searching, deleting or retrieving of files is performed.
6. When the option to retrieve files in ascending order is selected, user is displayed with ascending order.
7. The application is designed with modularity in mind. Even if one wants to update the path, they can change it through the source code. Application has been developed keeping in mind that there should be very less “hardcoding” of data.

## Conclusions

Further enhancements to the application can be made which may include:

* Conditions to check if user is allowed to delete the file or add the file at the specific locations.
* Handle the all-possible wrong input. Application will run steadily.
* Allowing user to append data to the file.