**Virtual key for repositories**

The project is developed by Kalviyarasi M.

The source code for the project is hosted at <https://github.com/kalviyarasi/Practise_project>

**Sprint Planning**

Number of sprints used to complete this project is 5

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint No | Sprint duration (in days) | Objective | Task Achieved |
| 1 | 1 | To create flow of the application. | Class, Main method and overall outline for the code is created |
| 2 | 3 | To create Add and Display method | Implementation of AddTheFile(), DisplayTheFile() methods by making use of path. |
| 3 | 3 | To create Delete and Search method | DeleteTheFile(), SearchTheFile() method has been implemented |
| 4 | 3 | To integrate all class using object creation | Two Switch cases were used inside the while loop to Integrate all the public class created |
| 5 | 2 | To test the code and its efficiency and to create specification document. | Source code is tested for various cases, and Exception handling is used accordingly to reduce the errors. |

**4.Algorithms**

Step 1: Create a java project and package in eclipse.

Step 2: Create class, main method and import the necessary libraries.

Step 3: Create separate public class and method to add a new file

Step 4: Attach your path in step 2;

Step 5: Repeat step 2 and step 3 to create class and methods to delete a file, search a file and to display the file.

Step 6: Create a while loop and get the user choice inorder to enter the switch case and perform operations like display and exit.

Step 7: Create a sub switch case to perform operations like add, delete and display files.

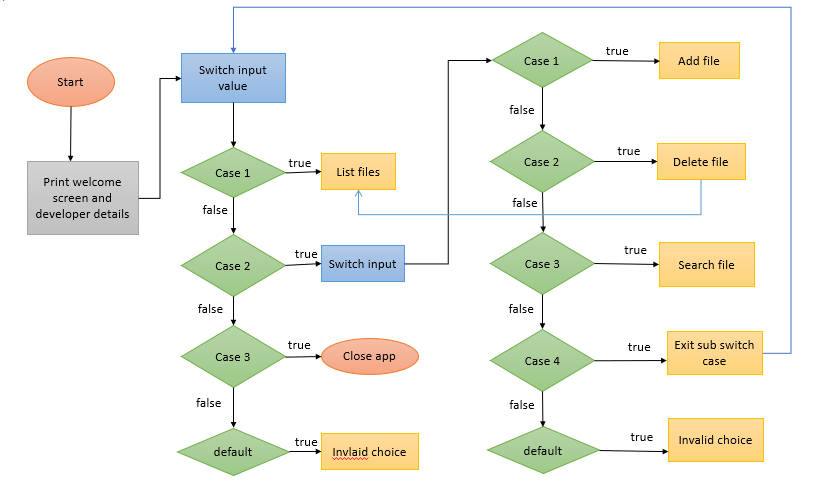
Step 8: Repeat step 6 until user exits from sub switch case.

Step 9: Repeat step 6 until user exits from the application.

Step 10: End the program.

**FlowChart**

A flow chart is a type of diagram that represents a workflow or process. A flow chart is a diagrammatic representation of an algorithm.



**Core Concepts**

**File Handling** is an important part of any application. File Handling is an integral part of any programming language as file handling enables us to store the output of any particular program in a file and allows us to perform certain operations on it. Java has several methods for creating, reading, updating and deleting files. The **File** classfrom **java.io** package, allows to work with files. The File class can be used by creating object of the class and then specifying the name of the file. The File class has many useful methods for creating and getting information about files.

For example,

createNewFile() creates an empty file and its return type is Boolean.

delete() deletes a file.

getName() returns the name of the file.

length() returns the size of the file in bytes.

**Collections framework** is a set of classes and interfaces that implement commonly resuable collection data structures. Although referred to as a framework, it works in a manner of a library.

**Sorting** is a waytoarrange elements of a list or array in a certain order. The order may be ascending or descending order. **Arrays,sort()** method used to sort array in ascending order.

**Exception Handling** is a mechanism to handle runtime errors. Exception is an unwanted or unexpected event, which occurs during the execution of the program. The keywords **try, catch, finally, throw and throws** form the base of exception handling.

**Recursion** adds clarity and reduces time needed to write and debug code. It also reduces time complexity. Recursion has more expressive power than iterative looping constructs.

**Flow Control** let you control the flow of the execution of the code in your program. In java flow control is achieved by placing decision making, branching, looping and adding conditional blocks.

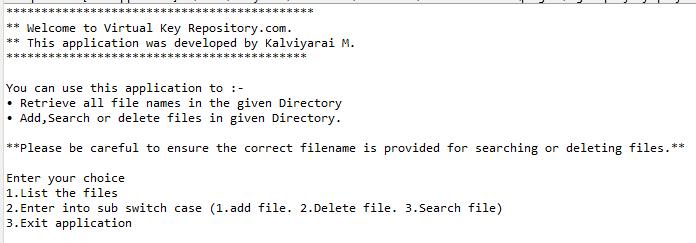
**Demonstarting the product capabilities, appearance and user interactions.**

**1.Welcome Screen**

A welcome window screen which will display the outline of all features incorporated in the project. It also displays the developer details in welcome window screen.

**Purpose:** This suggests the uneducated people or who is not familiar with this application, to proceed further by displaying all the available features.

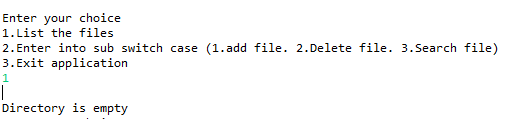
**Output screenshot**



**2.Displaying the files**

The user can view all the files that are stored in the directory. If there is no file in the directory it displays “Directory is empty”.

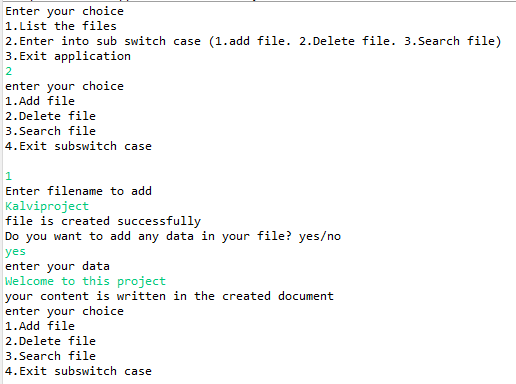
**Output Screenshot**

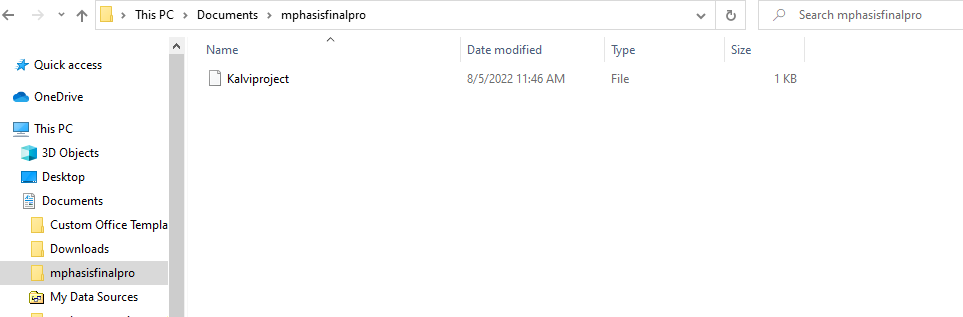


**3.Adding files to the directory**

The filename entered by the user is added to directory in the ascending order. In addition to this if user wants to add any data to the file, then user can add the data, after successful file creation.

**Output Screenshots**

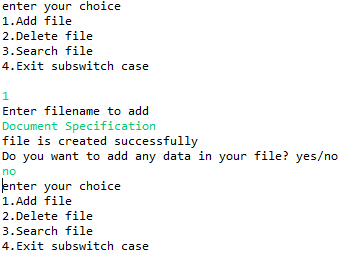


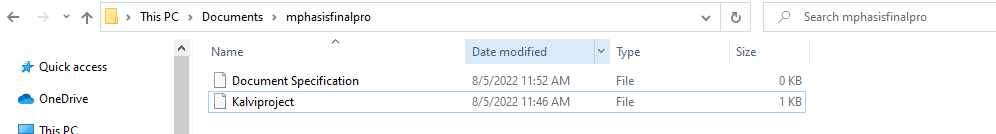




The above screenshot shows the data that is written in the file name “Kalviproject”

If the user doesn’t want to add any data to the file, it just create a file and add it to the directory.

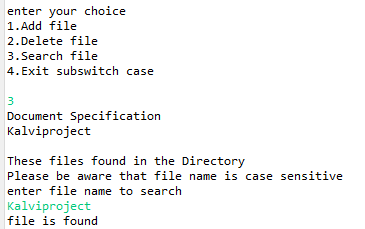


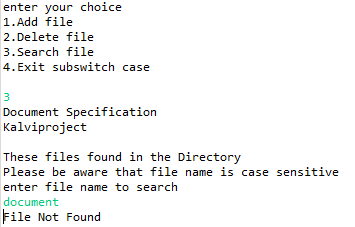


**3.Searching a file from directory**

The filename entered by the user is searched and if the file is found it returns “file is found” otherwise it returns “file not found”. This operation is case sensitive.

**Output screenshots**



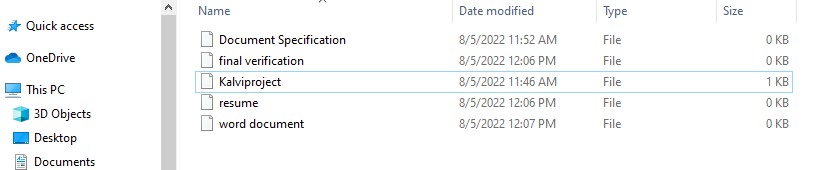


**4.Deleting file from directory**

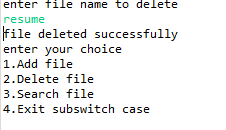
The user entered file is deleted from the directory. The file name is case sensitive so care should be taken before deleting the file.

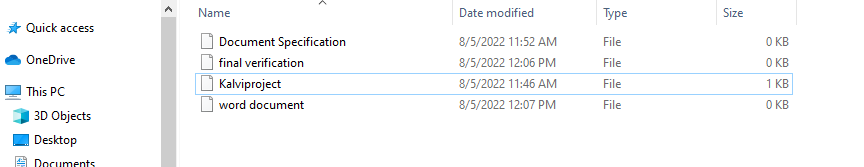
**Output screenshots**

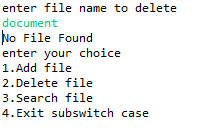
=>Before deleting file “resume”



=>After deleting file “resume”



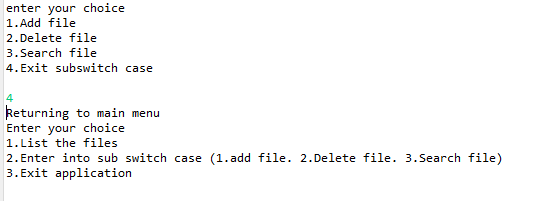




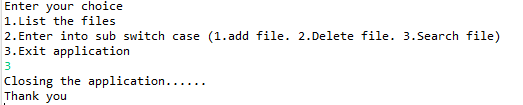
**5.Returning to the main menu**

This option is used for exiting from sub switch case and returns to the main menu.

**Output screenshot**



**6.Close the application**



These are the features that were added to this application. This application can be used to add, delete, search and display files from the directory.

**Pushing the code to github repository**

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

**cd <folder path>**

* Initialize the repository using the following command.

**git init**

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

**git add .**

* 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 push -u origin master**

using these commands code is pushed to the Github.

**Unique Selling Points of the application**

* The user is able to seamlessly switch between operations or returns to the previous menu even after any required operation like adding, searching, deleting or retrieving of files is performed
* When the user wants to delete any files, it displays the list of files in the directory so that user need not to remember the names of the file in the directory.
* The application is designed with modularity in mind. Even if one wants to update path, they can change it through the source code. Application has been designed in a way that there should be very less hardcoding of the data.
* The application is user friendly as it display all the operations that are available in the application to the users so that they can easily enter the choice.
* User is also provided with option to write the content if needed.
* The application is developed in such way that, it runs continuous by taking user inputs even when exception occurs.
* To terminate the application, appropriate option needs to be selected.

**Conclusions**

* Further enhancements to the application can be made which may include:
* Asking the user to add, delete or search file in different locations.
* Allowing users to append data to the file or rewrite the data.
* Retrieving files or folders by different criteria like Last Modified, type, size, etc.