**VIRTUAL KEY FOR YOUR REPOSITORY**

**SPECIFICATION DOCUMENT:**

This document contains the following details:

* Problem statement
* Developer details
* Algorithm and flowchart
* JAVA concepts used
* Sprints and GitHub link
* Unique Selling Points and Conclusion

**PROBLEM STATEMENT:**

To develop a prototype of the application – LOCKEDME.COM that contains features such as:

* Retrieving the files from a directory in ascending order
* Business level operations:
* Option to add a user specified file to the application
* Option to delete a user specified file to the application
* Option to search a user specified file to the application
* Navigation option to close the current execution context and return to the main context
* Option to close the application

**DEVELOPER DETAILS:**

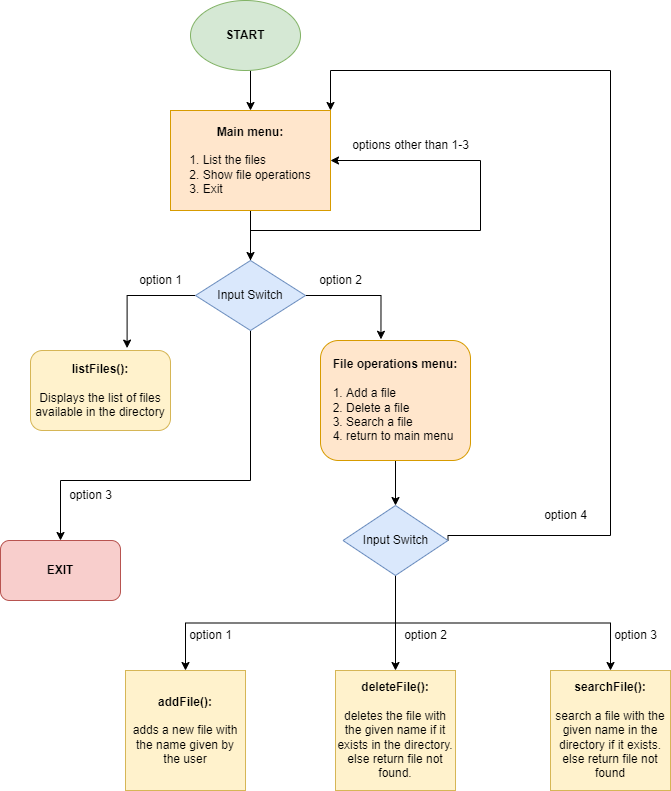
Name: Kamali S

GitHub: https://github.com/KamaliSrinivasan16

**ALGORITHM:**

* Print the introduction about the “LOCKEDME.COM” application and give the developer details and the main menu of the application appears.
* Option 1: List the files – It displays the list of files that are available in the mentioned directory.
* Option 2: Show file operations – It display the list of operations that can be performed on the files.
* Option 3: Quit – It is used to exit from the application.
* The choice is given by the user and the action is taken accordingly.
* Once option 1 is entered, the list of files will be displayed.
* If option 2 is entered, the file operation gets displayed. The options are:
  + - Add a file
    - Delete a file
    - Search a file
    - Return to menu
* If option 1 of file operations is entered by the user, ask for the new file name. A file with the new name is created in the same directory.
* If option 2 of file operations is entered by the user, ask for the file name. The file with the given name, if existed, is deleted from the directory, else returns file not found.
* If option 3 of file operations is entered by the user, ask for the file name. If a file with the given name already exists in the directory, acknowledge else returns file not found
* If option 4 of file operations is entered by the user, then display the main menu.
* If user enters option 3 of main menu, then exit from the application leaving a thanking note.
* If any other options are entered, return invalid option.

**FLOWCHART:**

****

**JAVA CONCEPTS USED:**

* OOPs
* Collections
* Exceptional handling
* Sorting
* Searching – linear search
* File handling

**SPRINTS:**

**DAY 1 Activities:**

* Creating directory with few files to perform file operations
* Created classes such as virtualKeyApp and welcomeScreen
* Created functions to display the welcome screen with application and developer details
* Created methods to list the files in the directory

**DAY 2 Activities:**

* Created a class named fileOperation to display the file operation menu
* Created methods to add, delete and search files in the directory
* Created try and catch statements to handle the exceptions
* Commented the entire for the ease of understandability
* Testing the entire program with different user inputs to ensure whether the requirements are satisfied or not.
* Pushed the code into GitHub to track the versions of the code
* Creating this specification document

**GITHUB LINK:**

<https://github.com/KamaliSrinivasan16/Virtual-Key-for-Your-Repositories/tree/main/src/virtualKeyApp>

**UNIQUE SELLING POINTS:**

1. The files are retrieved in ascending order that helps the user to identify the files easily.
2. The application is enabled to run without interruption as it is capable of handling the exceptions
3. The list of files is displayed after each file operations (add, delete) to ensure the user that whether the operation is performed or not
4. The user can easily switch from file operation menu to the main menu
5. User can provide options according to the menu. If entered options are incorrect, user gets another chance to enter the option with a warning.
6. User can exit from the application anytime by selecting the appropriate option.

**CONCLUSION**:

The features of the application can further be enhanced by adding:

* Options to read and write the files in the directory
* To append data to the file
* To display similar names of the files that matches partially with the file name entered to search
* Retrieving files based on recently modified date, type of files etc