LockedMe Application Product and Technical Specification document.

LockedMe.com

Kartikaya Srivastav

Source code : https://github.com/ksrivastav/LockedMe-Application.git

v (0.1)

Author: Kartikaya Srivastav

LockedMe Application Product
and Technical Specification
document.

1. Introduction

1.1. Purpose

1.2. Requirement Overview

Company Lockers Pvt. Ltd. hired you as a Full Stack Developer. They aim to digitize their products and chose LockedMe.com as their first project to start with. There is a requirement to develop a prototype of the application. The prototype of the application will be then presented to the relevant stakeholders for budget approval.

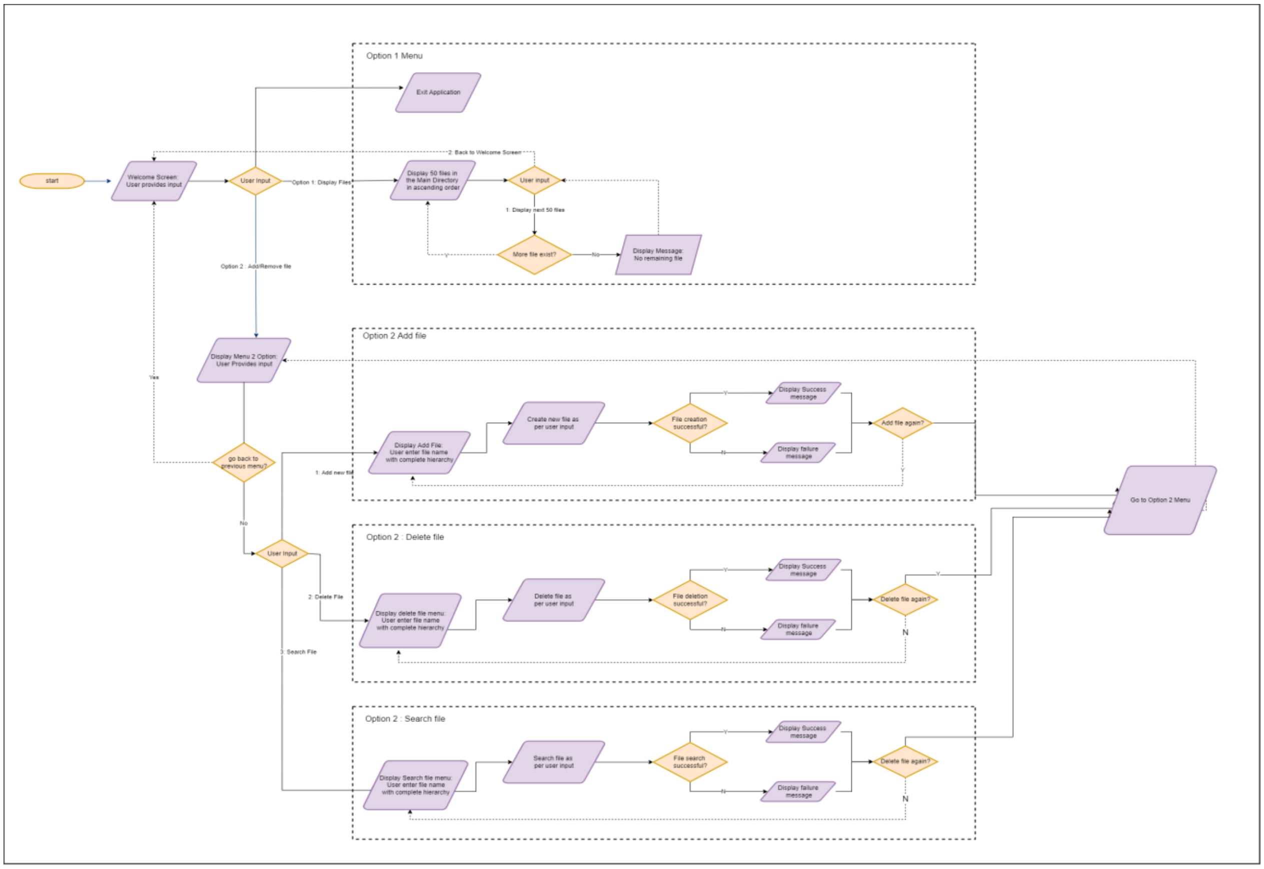
2. Product Specifications.

2.1. Product Specifications

Locked Me.com will be a command line-based application which will allow users to do the following:

I. Display the files in a specified directory.
II. Add new files in a specified directory.
III. Remove files from specified directory.

2.2. Applicaiton Flow Diagram

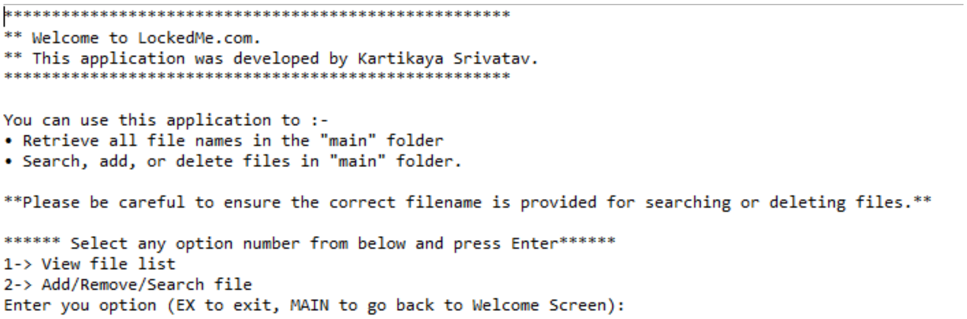


Author: Kartikaya Srivastav

2.3. User experience and user manual.

2.3.1. LockedMe prototype is a command-based application where users can perform various file handling operations using simple and easy to use commands.

2.3.2. When the Application starts, you are landed on the welcome screen. Below is the screenshot of the welcome screen.



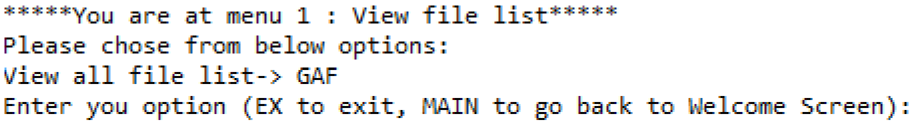
```
*****************************************************
** Welcome to LockedMe.com.
** This application was developed by Kartikaya Srivatav.
*****************************************************

You can use this application to :-
• Retrieve all file names in the "main" folder
• Search, add, or delete files in "main" folder.

**Please be careful to ensure the correct filename is provided for searching or deleting files.**

****** Select any option number from below and press Enter******
1-> View file list
2-> Add/Remove/Search file
Enter you option (EX to exit, MAIN to go back to Welcome Screen):
```

There are two menus in LockedMe application. Users can choose from the menu simply by entering the menu number of their choice.

- For example, if the user wants to navigate to menu 1: View file list. Simple press 1 and press enter.
- If user wants to exit the application at any time, simply enter “EX” command.
- If user wants to navigate to main menu at any point, simply enter ‘MAIN” command.

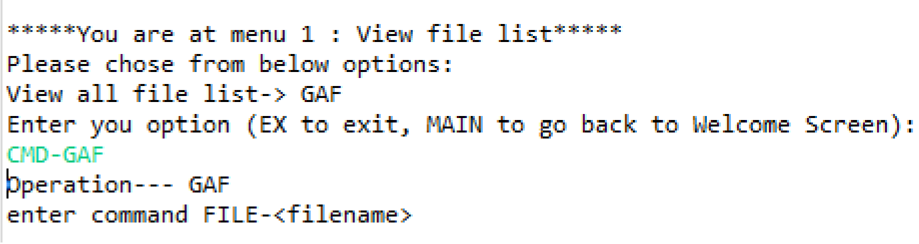
2.3.3. Working with Menu 1 (View file list)

2.3.3.1. After navigating to menu 1 from the main screen you will be displayed with a list of commands you can perform in menu 1.



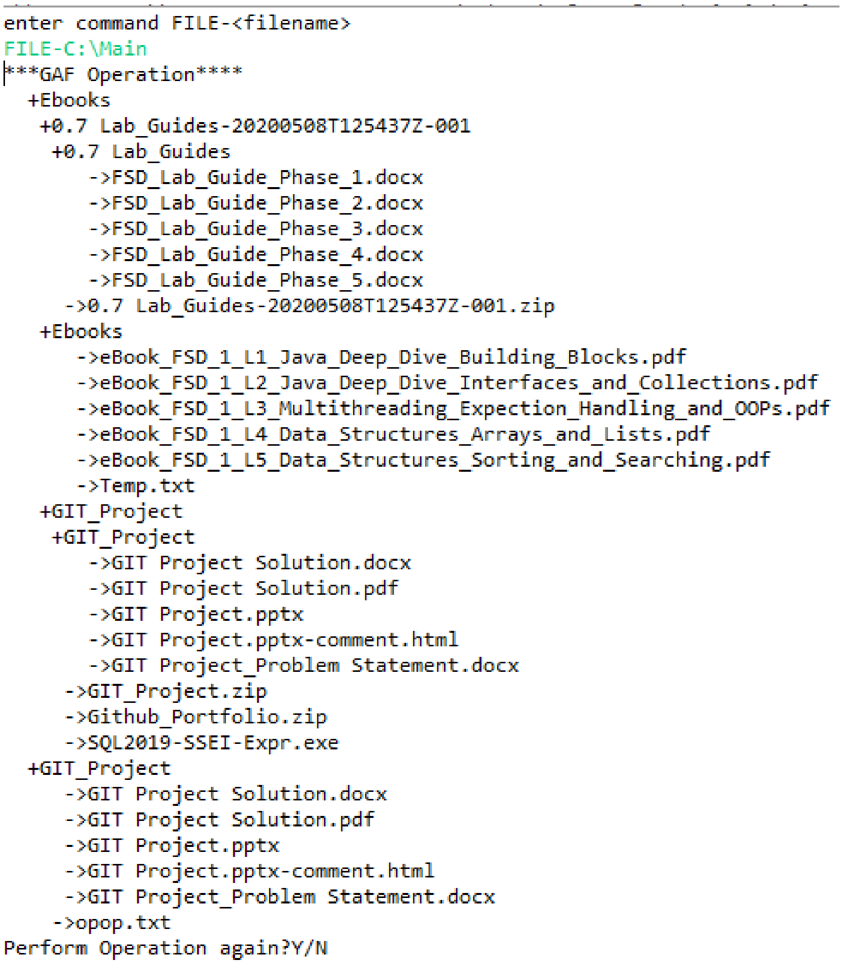
```
*****You are at menu 1 : View file list*****
Please chose from below options:
View all file list-> GAF
Enter you option (EX to exit, MAIN to go back to Welcome Screen):
```

2.3.3.2. Enter Command “CMD-GAF” to perform view all file list operation. With this operation, you can view all files in a directory including files in sub directories.



```
*****You are at menu 1 : View file list*****
Please chose from below options:
View all file list-> GAF
Enter you option (EX to exit, MAIN to go back to Welcome Screen):
CMD-GAF
Operation--- GAF
enter command FILE-<filename>
```

2.3.3.3. Now enter the directory you want to explore by simply entering command “FILE-<directory path>. Example FILE- C:\Main



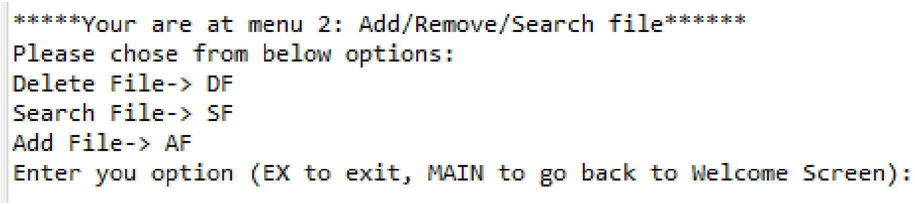
```
enter command FILE-<filename>
FILE-C:\Main
***GAF Operation****
  +Ebooks
   +0.7 Lab_Guides-20200508T125437Z-001
    +0.7 Lab_Guides
       ->FSD_Lab_Guide_Phase_1.docx
       ->FSD_Lab_Guide_Phase_2.docx
       ->FSD_Lab_Guide_Phase_3.docx
       ->FSD_Lab_Guide_Phase_4.docx
       ->FSD_Lab_Guide_Phase_5.docx
     ->0.7 Lab_Guides-20200508T125437Z-001.zip
   +Ebooks
      ->eBook_FSD_1_L1_Java_Deep_Dive_Building_Blocks.pdf
      ->eBook_FSD_1_L2_Java_Deep_Dive_Interfaces_and_Collections.pdf
      ->eBook_FSD_1_L3_Multithreading_Expection_Handling_and_OOPs.pdf
      ->eBook_FSD_1_L4_Data_Structures_Arrays_and_Lists.pdf
      ->eBook_FSD_1_L5_Data_Structures_Sorting_and_Searching.pdf
      ->Temp.txt
   +GIT_Project
    +GIT_Project
       ->GIT Project Solution.docx
       ->GIT Project Solution.pdf
       ->GIT Project.pptx
       ->GIT Project.pptx-comment.html
       ->GIT Project_Problem Statement.docx
     ->GIT_Project.zip
     ->Github_Portfolio.zip
     ->SQL2019-SSEI-Expr.exe
  +GIT_Project
     ->GIT Project Solution.docx
     ->GIT Project Solution.pdf
     ->GIT Project.pptx
     ->GIT Project.pptx-comment.html
     ->GIT Project_Problem Statement.docx
    ->opop.txt
Perform Operation again?Y/N
```

2.3.3.4. You can perform the same operation on different (or same) directory again by entering “Y” or you can go back to previous menu by entering “N”

2.3.3.5. To view all files in a directory (including files in the subdirectory).

2.3.4. Working with Menu 2(Add/Remove/Search file)

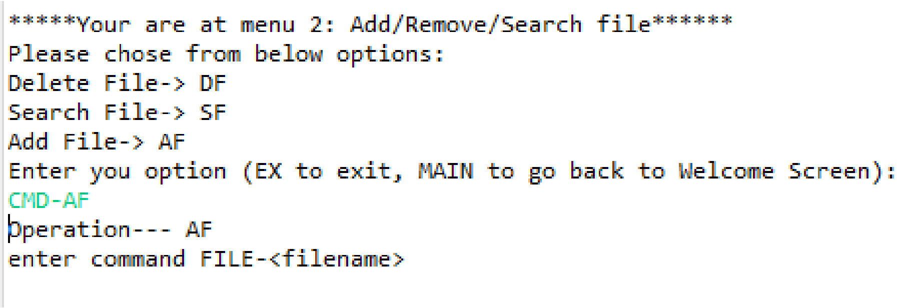
2.3.4.1. After navigating to menu 2 from the main screen you will be displayed with a list of commands you can perform in menu 2. To perform any operation, you need to enter the command using format CMD-<operation you want to perform>. for example, CMD-DF, if you want to perform delete file operation. Similarly, CMD-SF (to perform search a file operation) and CMD-AF to create a new file.



```
*****Your are at menu 2: Add/Remove/Search file******
Please chose from below options:
Delete File-> DF
Search File-> SF
Add File-> AF
Enter you option (EX to exit, MAIN to go back to Welcome Screen):
```

2.3.4.2. Creating a new File (CMD-AF)

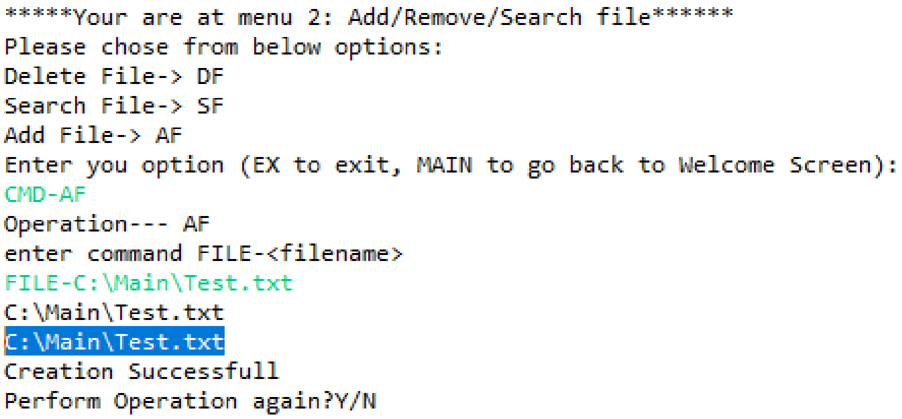
2.3.4.2.1. To create a new file, enter command CMD-AF. You will be asked for file path and file name on which you want to create the file.



```
*****Your are at menu 2: Add/Remove/Search file******
Please chose from below options:
Delete File-> DF
Search File-> SF
Add File-> AF
Enter you option (EX to exit, MAIN to go back to Welcome Screen):
CMD-AF
Operation--- AF
enter command FILE-<filename>
```

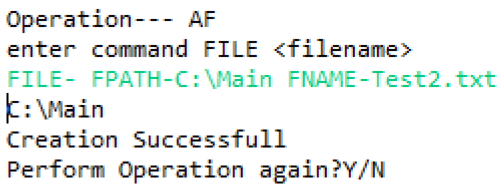
2.3.4.2.2. You can enter file name to be created, you must use FILE- command. You can use FILE- in two ways: -

2.3.4.2.2.1. You can either enter full path “FILE-C:\Main\Test.txt.”



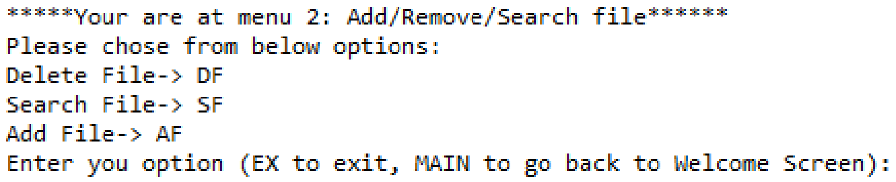
```
*****Your are at menu 2: Add/Remove/Search file******
Please chose from below options:
Delete File-> DF
Search File-> SF
Add File-> AF
Enter you option (EX to exit, MAIN to go back to Welcome Screen):
CMD-AF
Operation--- AF
enter command FILE-<filename>
FILE-C:\Main\Test.txt
C:\Main\Test.txt
C:\Main\Test.txt
Creation Successfull
Perform Operation again?Y/N
```

2.3.4.2.2.2. Or you can separately specify the file path and file name. Format is **FILE- FPATH-<directory path> FNAME-<name of the file>** For example:



```
Operation--- AF
enter command FILE <filename>
FILE- FPATH-C:\Main FNAME-Test2.txt
C:\Main
Creation Successfull
Perform Operation again?Y/N
```

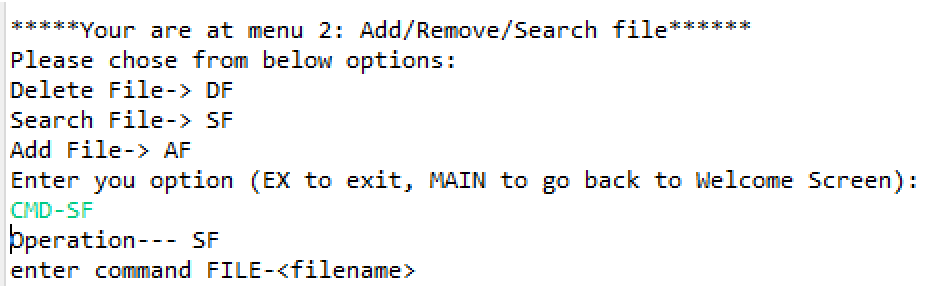
2.3.4.2.2.3. You can choose Y to create the new file again or you can choose N to go back to the previous menu



```
*****Your are at menu 2: Add/Remove/Search file******
Please chose from below options:
Delete File-> DF
Search File-> SF
Add File-> AF
Enter you option (EX to exit, MAIN to go back to Welcome Screen):
```

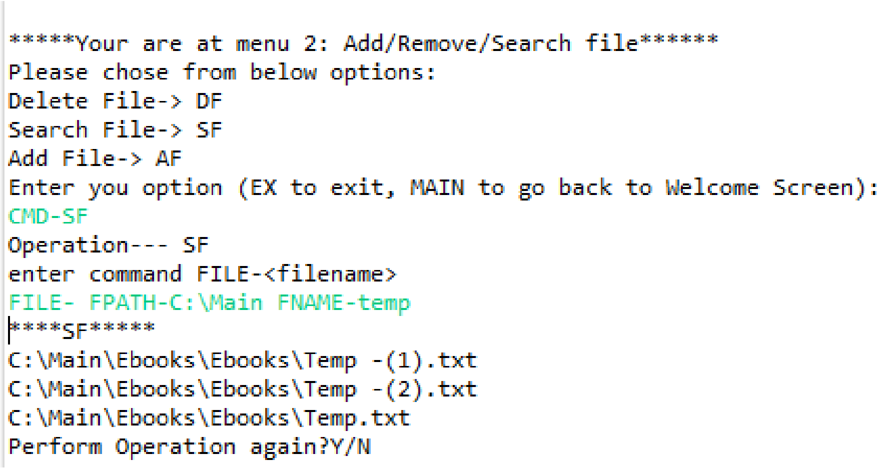
2.3.4.3. Search for a file (CMD-SF)

2.3.4.3.1. To search a file in a directory and its sub directories, enter command CMD-SF. You will be asked for file path and file name on which you want to create the file. **Please note that SF command allows you to perform fuzzy searches, thus, to search for a file you need not specify the exact path and exact name of the file.**



```
*****Your are at menu 2: Add/Remove/Search file******
Please chose from below options:
Delete File-> DF
Search File-> SF
Add File-> AF
Enter you option (EX to exit, MAIN to go back to Welcome Screen):
CMD-SF
Operation--- SF
enter command FILE-<filename>
```

2.3.4.3.2. Enter the file directory and file name you want to search for. The format for the command is **FILE- FPATH-<directory path> FNAME-<name of the file>**. For example: FILE-FPATH-C:\Main FNAME-temp

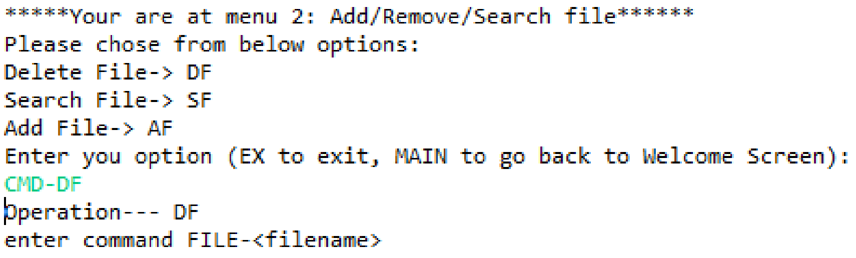


```
*****Your are at menu 2: Add/Remove/Search file******
Please chose from below options:
Delete File-> DF
Search File-> SF
Add File-> AF
Enter you option (EX to exit, MAIN to go back to Welcome Screen):
CMD-SF
Operation--- SF
enter command FILE-<filename>
FILE- FPATH-C:\Main FNAME-temp
****SF*****
C:\Main\Ebooks\Ebooks\Temp -(1).txt
C:\Main\Ebooks\Ebooks\Temp -(2).txt
C:\Main\Ebooks\Ebooks\Temp.txt
Perform Operation again?Y/N
```

2.3.4.3.3. Choose Y/N to perform the same operation again or go back to previous menu.

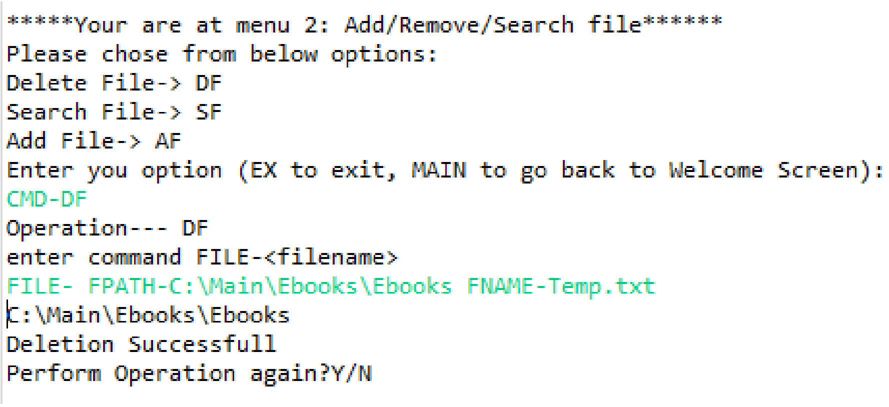
2.3.4.4. Delete a file (CMD-DF).

2.3.4.4.1. To delete a file in a directory, enter command CMD-DF. You will be asked for the file path and file name on which you want to delete the file.



```
*****Your are at menu 2: Add/Remove/Search file******
Please chose from below options:
Delete File-> DF
Search File-> SF
Add File-> AF
Enter you option (EX to exit, MAIN to go back to Welcome Screen):
CMD-DF
Operation--- DF
enter command FILE-<filename>
```

2.3.4.4.2. Enter the file path and file name of the file you want to delete. (exact path and exact name only).

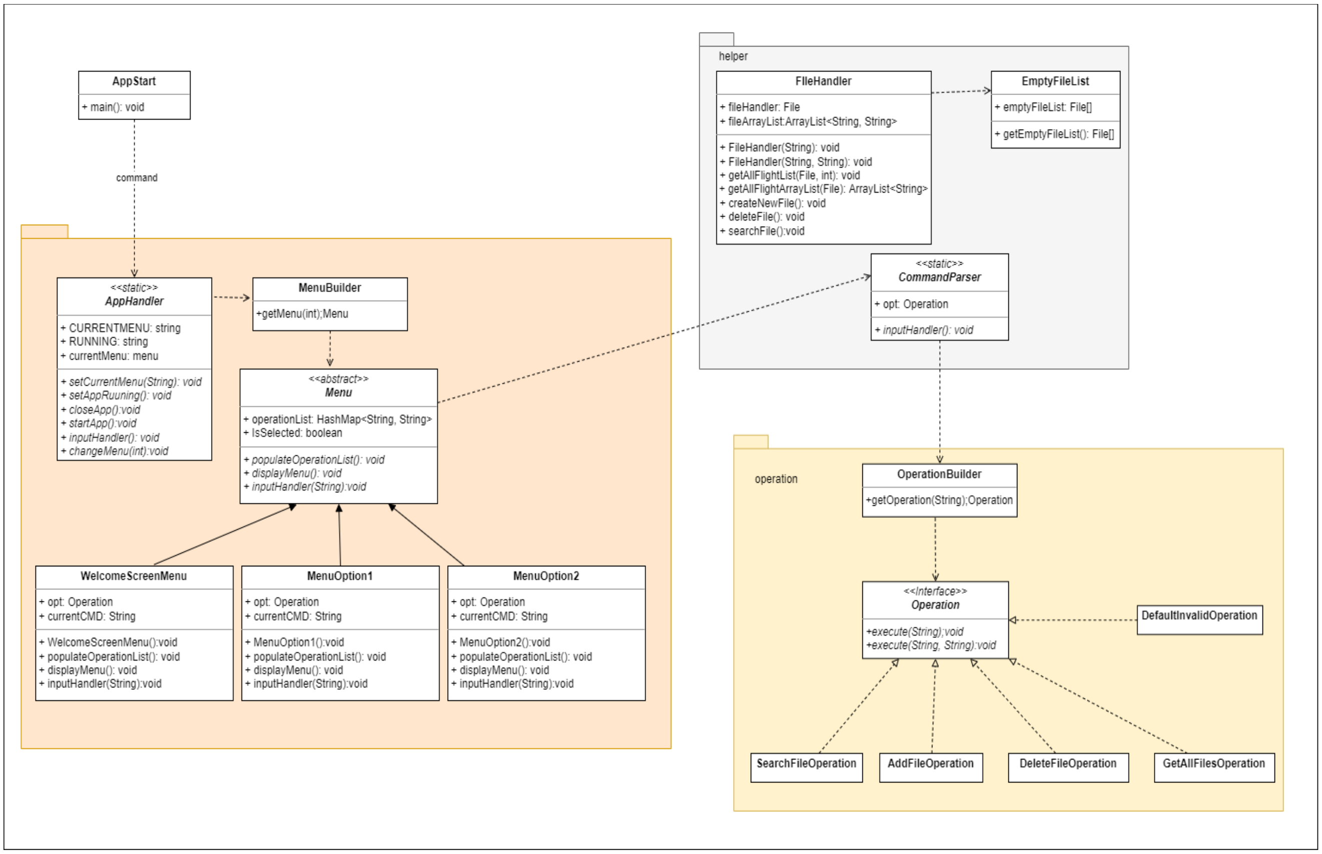


```
*****Your are at menu 2: Add/Remove/Search file******
Please chose from below options:
Delete File-> DF
Search File-> SF
Add File-> AF
Enter you option (EX to exit, MAIN to go back to Welcome Screen):
CMD-DF
Operation--- DF
enter command FILE-<filename>
FILE- FPATH-C:\Main\Ebooks\Ebooks FNAME-Temp.txt
C:\Main\Ebooks\Ebooks
Deletion Successfull
Perform Operation again?Y/N
```

2.3.4.4.3. Choose Y/N to perform the same operation again or go back to previous menu.

3. System Design

3.1. UML Diagram



Package Summary:

- **Menu:** The package that included all classes and abstract classes that builds Menu options of the application including WelcomeScreenMenu and MenuOption1, MenuOption2. This package implements Factory Design Pattern to help in building scalable menu options system.
- **Operations:** this package includes all the classes and interfaces that is used to implements all the file operations that user can perform on files including (GAF, AF, SF and DF). This package works on Strategy Design Pattern
- **Helper:** this package contains various classes and miscellaneous files that implement common functionalities used at various levels in the application. This package helps other packages in performing their common tasks.

3.2. Application Technical Specifications:

Application has been designed to stick to SOLID principles. Application is designed to be easily scalable where new functionalities can be added without impacting existing ones. The following points have been taken into consideration while designing the application:

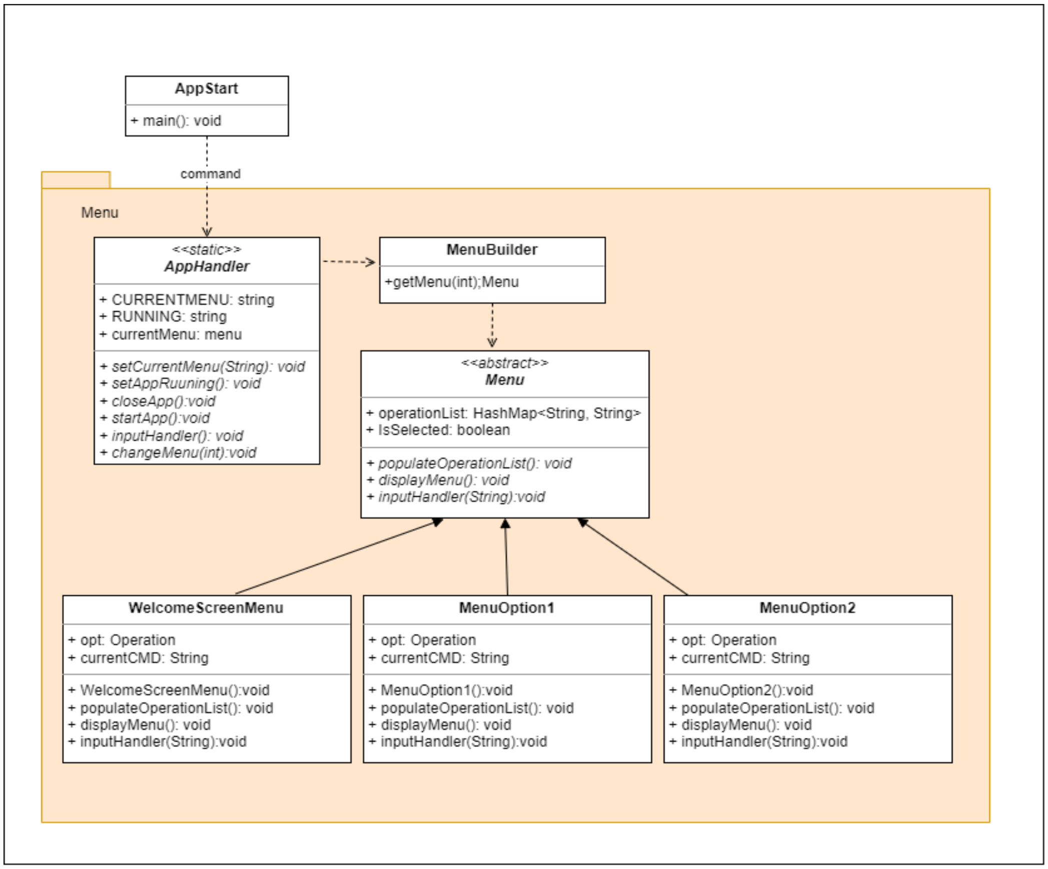
- ➢ Application should be easily scalable.
- ➢ Application should have the capability to be added with more menu options easily without impacting existing menu options.
- ➢ Individual menu options can be added with new operations easily without impacting the existing ones.
- ➢ High code reusability but with maximum loose coupling.
- ➢ Thus the application should be maximum closed for modification and open for extension.

3.2.1. Design Pattern Used:

3.2.1.1. Factory Pattern.

3.2.1.1.1. Menu options in the applications are invoked using Factory Method pattern. The menu package contains all the classes related to menu options. Individual Menu options(WelcomeScreenMenu, MenuOption1, MenuOption2) extends the Menu abstract class.

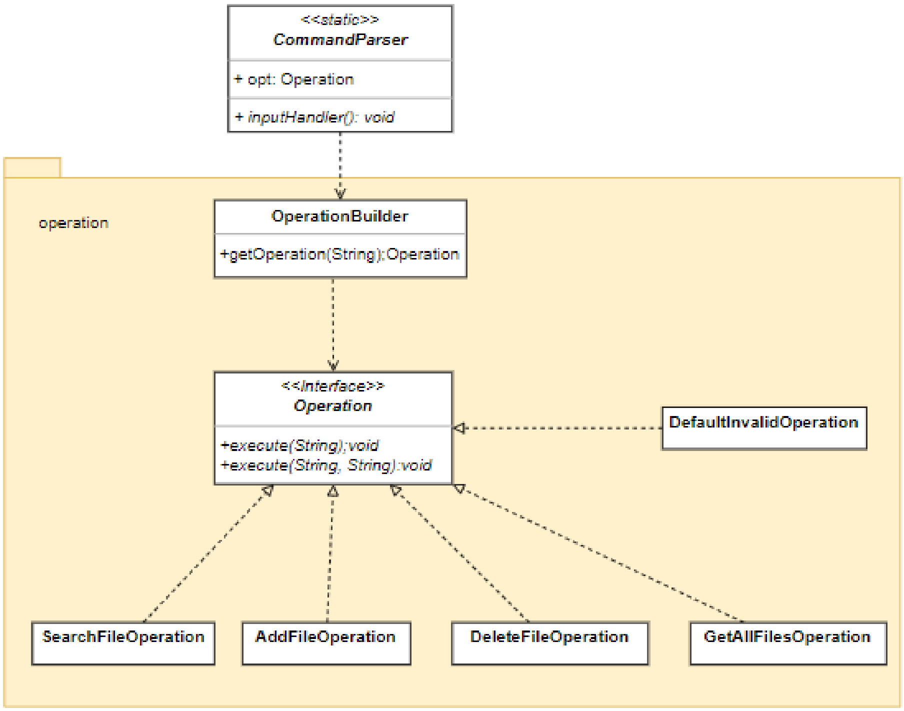
3.2.1.1.2. Instead of invoking MenuOptions Individually from AppHandler class. The responsibility of returning and required correct MenuOption is with MenuBuilder. The AppHandler simply calls MenuBuilder for the required menu.



3.2.1.1.3.

3.2.1.2. Strategy Pattern Design

3.2.1.2.1. Operation package contains all the classes required to build operations that user can perform on files using this application.



3.2.1.2.2. Any new operation that needs to be included in the application can be simply added by implementing operation interface and plugging it into OperationBuilder.

3.2.1.2.3. CommandParser which is used to interpret commands from user will simply pass the enter command to OperationBuilder which as per the command entered invokes the required operation.

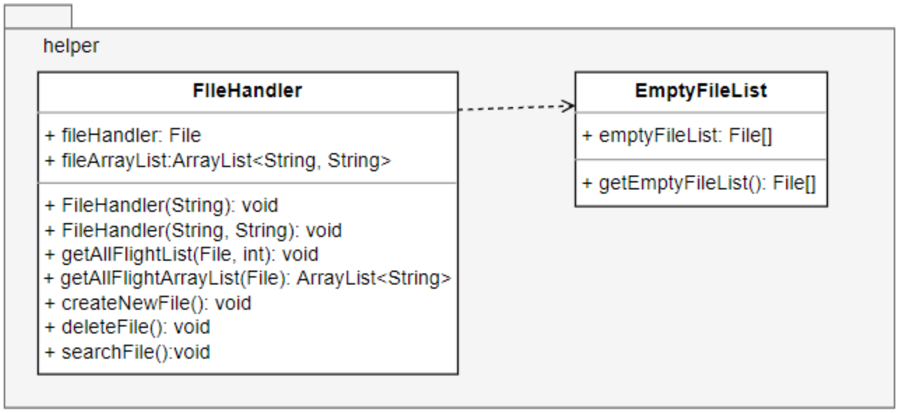
3.2.1.3. NULL object Reference method.

The application makes use of NULL object reference pattern so that no special code is required to handle instances in which an operation either yields no result or user enters a command which is not supported by application.

3.2.1.3.1. Null object reference pattern has been implemented at two levels.

3.2.1.3.1.1. Command Level: application can handle any invalid command that user enters.

3.2.1.3.1.2. Operation Level: application can display appropriate message without any special implementation. If an operation ends up with no result, EmptyFileHandler is used to return a NoFileFound message.


helper
FileHandler
+ fileHandler: File
+ fileArrayList:ArrayList<String, String>
+ FileHandler(String): void
+ FileHandler(String, String): void
+ getAllFlightList(File, int): void
+ getAllFlightArrayList(File): ArrayList<String>
+ createNewFile(): void
+ deleteFile(): void
+ searchFile():void
EmptyFileList
+ emptyFileList: File[]
+ getEmptyFileList(): File[]

3.2.2. Data structures used

3.2.2.1. ArrayList: To perform handle collection of files.

3.2.2.2. HashMap: maps command with operations.

4. Sprint Planning:

The Application has been implemented in two sprints

Sprint Number	Days	Functionalities Delivered
1	15	• Operations package • Helper package
2	15	• Menu package • Application integration and system testing

5. Unique Selling Points:

5.1. Application is extremely easy to use.

5.2. Command based implementation using various design pattern to support a wide variety of operations.

5.3. Highly scalable application.

5.4. Easier file searching operation using fuzzy searches with directory and sub directory scanning.

5.5. Robust list files operation with directory and subdirectory scanning and user-friendly hierarchal display of data.

6. Further improvements:

6.1. Relieve user from specifying exact file name for certain operations.

6.2. Allow users to append files.

6.3. Integration of resource file to configure repetitive config and data.

7. GIT Repository
   7.1. Link of repository : https://github.com/ksrivastav/LockedMe-Application.git
   7.2. To create repository on command line :

Link:
```
echo "# LockedMe-Application" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/ksrivastav/LockedMe-Application.git
git push -u origin main
```

7.3. Push an existing repository from the command line

```
git remote add origin https://github.com/ksrivastav/LockedMe-Application.git
git branch -M main
git push -u origin main
```