# LockedMe – Virtual Key for Repositories
# Developer: Kamar Zaghloul

## 1. Info

- FSD 2021 (Vodafone)
- **Developer**: Kamar Zaghloul
- **GitHub Repo**: https://github.com/kamarz01/LockedMe_Application_Phase1

## 2. Introduction

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. You're asked to develop a prototype of the application. The prototype of the application will be then presented to the relevant stakeholders for the budget approval. Your manager has set up a meeting where you're asked to present the following in the next 15 working days (3 weeks):

- Specification document - Product's capabilities, appearance, and user interactions
- Number and duration of sprints required
- Setting up Git and GitHub account to store and track your enhancements of the prototype
- Java concepts being used in the project
- Data Structures where sorting and searching techniques are used.
- Generic features and three operations:
- Retrieving the file names in an ascending order
- Business-level operations:
- Option to add a user specified file to the application
- Option to delete a user specified file from the application
- Option to search a user specified file from the application
- Navigation option to close the current execution context and return to the main context

- Option to close the application

## 3. Project Planning

- The project is to be completed in two sprints.
- The resources available for this project is (2): 1 Developer & 1 Tester.

Sprint (1) will include:

- Create application design.
- Create project skeleton.
- Creating the Welcome screen.
- Implement List all files feature.
- Implement Close app feature.
- Implement the More options screen.
- Implement Add new files feature.

Sprint (2) will include:

- Implement Deleting files
- Implement Searching for files
- Improve exception handling
- Improve negative inputs handling
- Testing the whole application.
- Creating documentations.
- Pushing project to GitHub.

## 4. Java concepts used in Project

- Exception Handling
- OOP
- Recursion
- Collections
- File & IO
- Streams
- Enums
- Data Structures

  And more

## 5. Application Flow

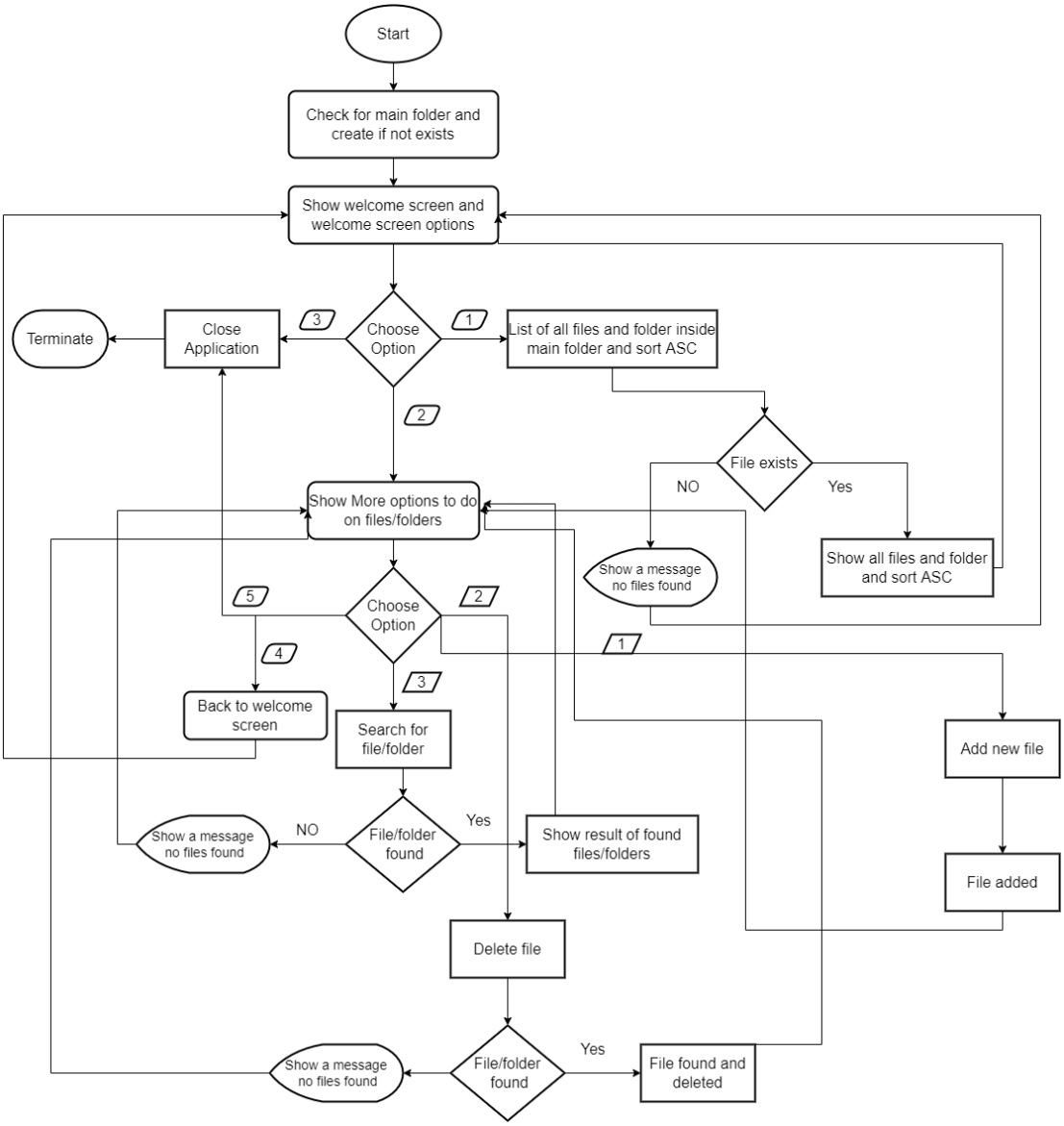


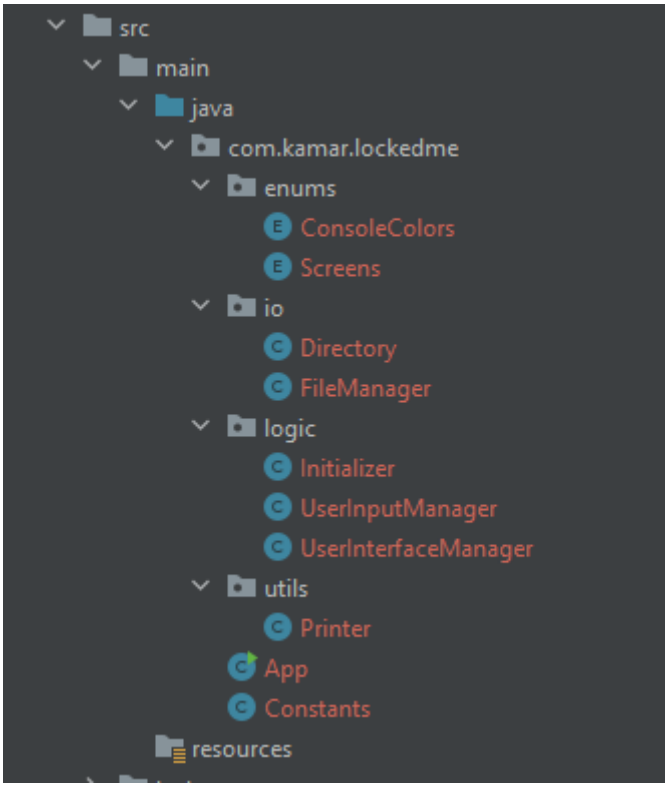

**Figure (1):** Application Flow

## 6. Project Structure

The project was been created following a good design structure to separate the logic from the user interface from file and directories handlers.

Some utlis were created specifically for this project, such as Coloring the Console output based on result.

The project structure could be found below:

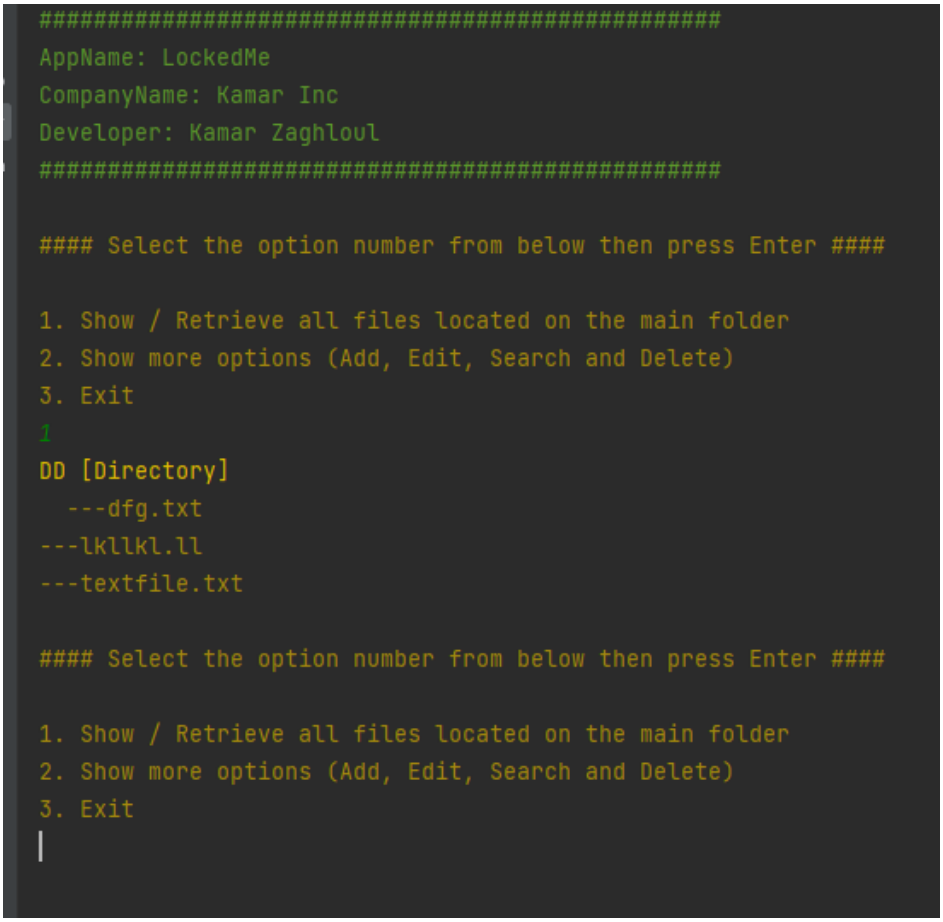



## 7. Application Demo

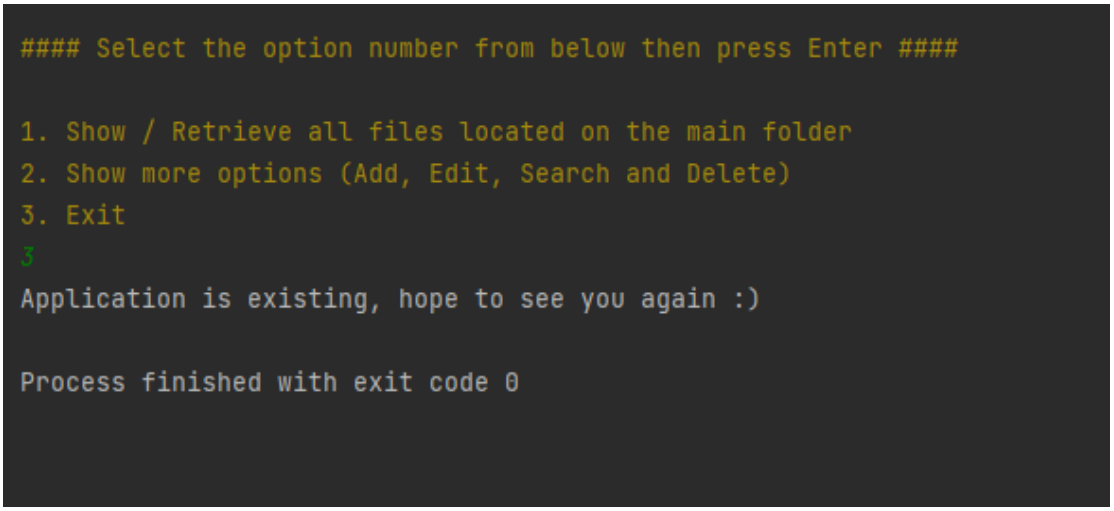
### Welcome Screen

```
##################################################
AppName: LockedMe
CompanyName: Kamar Inc
Developer: Kamar Zaghloul
##################################################

#### Select the option number from below then press Enter ####

1. Show / Retrieve all files located on the main folder
2. Show more options (Add, Edit, Search and Delete)
3. Exit
```

## After selecting option (1), listing all files ASC




## Closing Application



## Screen (2) more options

```
################################################
AppName: LockedMe
CompanyName: Kamar Inc
Developer: Kamar Zaghloul
################################################

#### Select the option number from below then press Enter ####

1. Show / Retrieve all files located on the main folder
2. Show more options (Add, Edit, Search and Delete)
3. Exit
2
#### Select the option number from below then press Enter ####

1. Add a new file.
2. Delete an existing file/folder.
3. Search for a file.
4. Back to Welcome screen
5. Exit
```

## Adding New File

```
#### Select the option number from below then press Enter ####

1. Add a new file.
2. Delete an existing file/folder.
3. Search for a file.
4. Back to Welcome screen
5. Exit
1
Please type the file name you would like to create:
Testfile.txt
File Testfile.txt created successfully
#### Select the option number from below then press Enter ####

1. Add a new file.
2. Delete an existing file/folder.
3. Search for a file.
4. Back to Welcome screen
5. Exit
```

## Deleting Files

```
#### Select the option number from below then press Enter ####

1. Add a new file.
2. Delete an existing file/folder.
3. Search for a file.
4. Back to Welcome screen
5. Exit
2
Please type the name of the file/folder to be deleted:
Testfile.txt
File Testfile.txt found and deleted successfully
#### Select the option number from below then press Enter ####

1. Add a new file.
2. Delete an existing file/folder.
3. Search for a file.
4. Back to Welcome screen
5. Exit
```

## Searching for Files

```
#### Select the option number from below then press Enter ####

1. Add a new file.
2. Delete an existing file/folder.
3. Search for a file.
4. Back to Welcome screen
5. Exit
3
Please type the name of the file/folder to search for:
1.txt
List of files found:
File 1.txt on Path: C:\Users\[redacted]n\Simplile
File 1.txt on Path: C:\Users\[redacted]n\Simplile
#### Select the option number from below then press Enter ####

1. Add a new file.
2. Delete an existing file/folder.
3. Search for a file.
4. Back to Welcome screen
5. Exit
```

```
Implement OOPS using JAVA with Data Structures and Beyond\8. Final Project\main_folder\1.txt
Implement OOPS using JAVA with Data Structures and Beyond\8. Final Project\main_folder\DD\1.txt
```

### Back to Welcome Screen

```
Please type the name of the file/folder to search for:
1.txt
List of files found:
File 1.txt on Path: C:\Users\Vodafone-IoT\Desktop\SimpliLearn\Simplilea
File 1.txt on Path: C:\Users\Vodafone-IoT\Desktop\SimpliLearn\Simplilea
#### Select the option number from below then press Enter ####

1. Add a new file.
2. Delete an existing file/folder.
3. Search for a file.
4. Back to Welcome screen
5. Exit
4

#### Select the option number from below then press Enter ####

1. Show / Retrieve all files located on the main folder
2. Show more options (Add, Edit, Search and Delete)
3. Exit
```

## 8. Application unique selling points

- The app handle all invalid events, exceptions and do not crash of close unexpectedly.
- Data structure & Algorithms concepts were used, so application performance will be great and efficient.
- User can create files and folders in main root or even in any nested path.
- The program searches recursively in all sub folders to get you the file / folder you search for.
- The application display the exact error the user did (permission error, not empty directory, not found and so on).
- Application logic is designed with high quality and modular.

9. **Github Repo**

   **https://github.com/kamarz01/LockedMe_Application_Phase1**