**Virtual Key for Repositories**

**1. Project Description**

Ass a Full Stack developer, Company Lockers Pvt. Ltd. has hired me. They want to digitalize their products and have chosen LockedMe.com as their first project to start with. They want a prototype of the application, and, for this, the user interaction will be via a command line.

This prototype must display a welcome screen with the Application name and the developer details. This application is used to store files in a root directory. For that purpose, it displays a menu with three options: the first one returns the current file names in ascending order; the root directory can be either empty or contain few files or folders in It. The second option leads to a submenu with four options: add a file, delete a file, search a file, return to main menu. The third option closes the application. The name or the files are non-case sensitive, so it does not matter If the user fails with upper and lower cases. This prototype represents my library, so book names are stored in the root directory.

The code has been pushed in this address:

<https://github.com/maritxusalas/MyProject>

My credentials to defend this project are. My name is Maria Salas, I am a Telecommunication Engineer and I have more than 25 years’ experience in a Telephony Company; usually I don't develop programming projects, but I want to start with it to improve my department efficiency, so this is a good first step to achieve It.

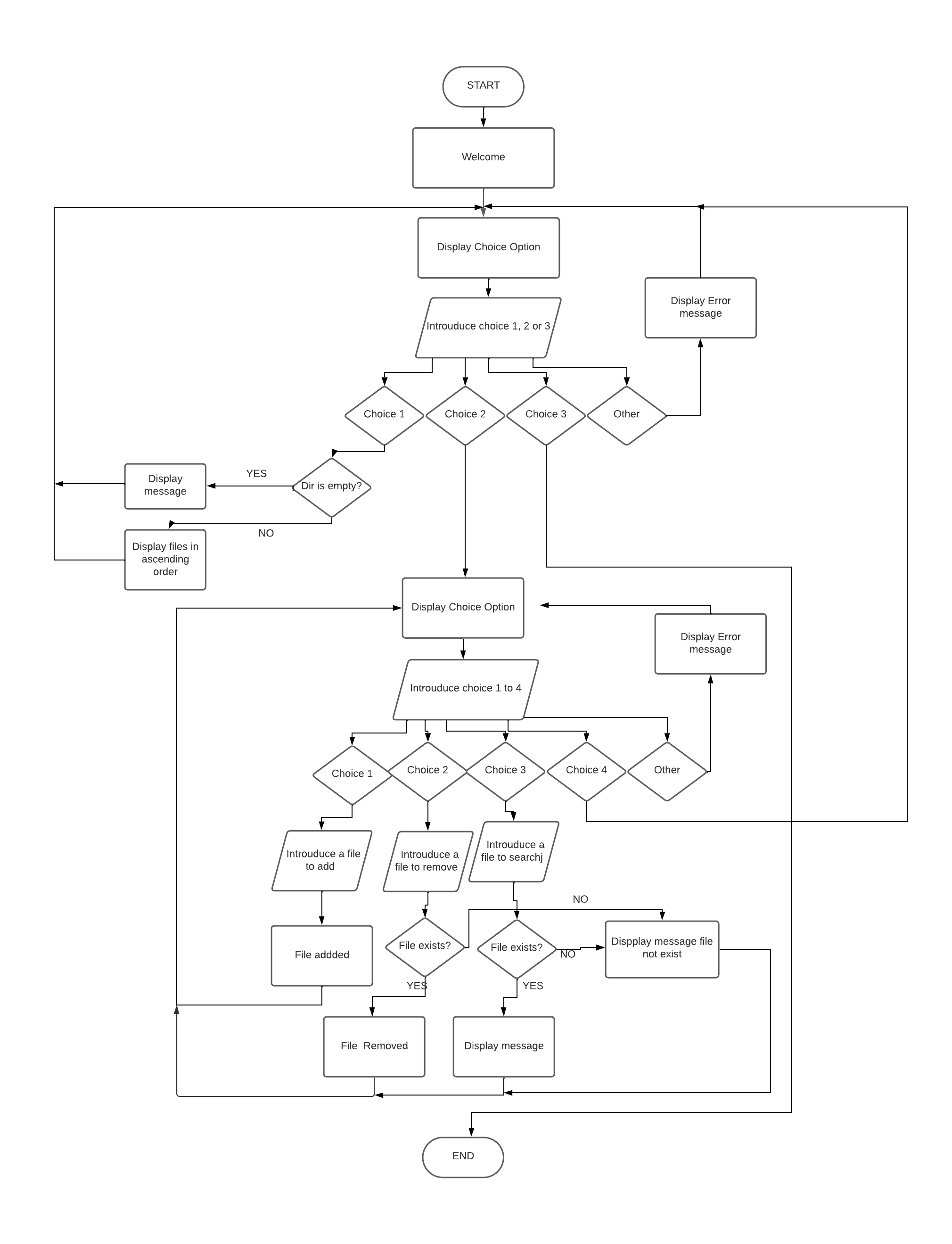
**2. Sprints planning**

1. Sprint 1: Planning

* Duration: 1 week
* Objectives
  + Requirement analysis
  + Flow chart design
  + Github and development environment configuration. Eclipse IDE has been chosen as the tool to make the prototype
  + Packages and classes creation
* Results
  + Requirements documentation
  + Flow chart designed
  + GitHub created
  + Root directory defined, Eclipse ID environment configured, and git result pushed to GitHub created

1. Sprint 2: code development
   * Duration: 1 week
   * Objectives
     + Develop the initial code that meet the application requirements. Very basic code
     + This code does not implement case sensitive differentiation or errors handling
     + Perform unit tests
   * Results
     + First prototype made
     + Unit tests satisfactory
2. Sprint 3: code debugging
   * Duration: 1 week
   * Objectives
     + Error handling developed in case options. Delete, add, search functions improved.
     + Stakeholders’ tests performed
     + Finish documentation
   * Results
     + Program finished with tests completed
     + Errors fixed
     + Documentation uploaded

**3. Application Flowchart**



**4. Project Documentation**

1. **Steps to use Git and push the project in the GitHub Repository**

* I initialize git in

C: \Users\msalas1\eclipse-workspace\VirtualKeyForRepositories\MyProject

Right Button. Abrir en Terminal

- git init

- git status

Now I must add the changes

-git add .

Now commit the changes

- git commit -m "…"

- git branch -M main

Añadimos el remote repository

- git remote add origin https://token@ github.com/maritxusalas/MyProject

Volcamos el proyecto

- gipt push -u origin main

1. **Core concepts used in the project**

* I have used 2 packages thoug no necessary. One to manage menus and interactions with users and the other one to define the different methods called by the main
* First one Is com.virtual.main
  + First class is Wellcome. Here you run the program.
    - * This class just shows the program presentation and calls de method MenuOptions located in MainMenu
  + SecondClass is MainMenu that presents the 3 options of main menu and manage exceptions. It uses the following concepts
    - * Use of scanner an BufferedReader classes to take values throug keyboard
      * Use while statement to go back to main menu but option 3
      * Use of handling exeption try-catch to handle an exception in the case that we introduce one option not integer
      * Use of choice statement to select the different options of the main menu
      * We have used a variable to call a method of a class
      * Use of if statement
  + Second class is SubMenu that has passing patameter method but nor return type. In the submenu it Is not used any other core concept.
* Second One Is com.virtual.file. We must import in In com.virtual.key this package. In this packages the different oprtions asked are defined in classes and methods in each class
  + The first class is File Service. This class has a method that returns a LinkedList wich contains the manes of the files stored in the root direcory defined in the main class
    1. The method used Is a method that passes parameters and return a LinkedList
    2. For this purpose we must import java.utlil.LInkedlist
    3. In this method we also use the library java.io.File. This Library allows to make operations with files and directories and tha is what we need in this project
    4. Using the properties of the library java.io.File we define a variable dir thart stores a reference to a class File Object which represents one directory in the file system
    5. Also create a files variable thar stores files and subdirs inside dir directoru
    6. Finally using the proprerties of class file, the names of the files are stored in Linkedlist variable Filenames
    7. In this method if and for statements are used
  + The second class Is SortFile. This class has a very simple methos
    1. The method SorFiles receive an string variable path whichh Is the directory and returns a variable LinkedList sorted
    2. Variable FileNames obtains its values from FileService class
    3. We use the property of LinkEdList Collections to sort the files In the list
  + Class AddFile. This class has one method thar receives two parameters: file to add and directory in wich store this file
    1. In this method java.io.File properties are also used: File dir, File file, file.exists(); FileOutpustStream.
    2. To manage FileOutputStream is necessary to introduce exception handling try/catch
    3. It is also used linkEdList properties such as add
  + Class DeleteFile. It has a method that removes a file form directory if it exists
    1. In this class It Is used a new porperty of LinkedList which is Iterator. This is an interface used to search the file to delete.
    2. It is also used the java.io.File delete property
  + Class SearchFile. It works the same as DeleteFile without removing it.

-

.