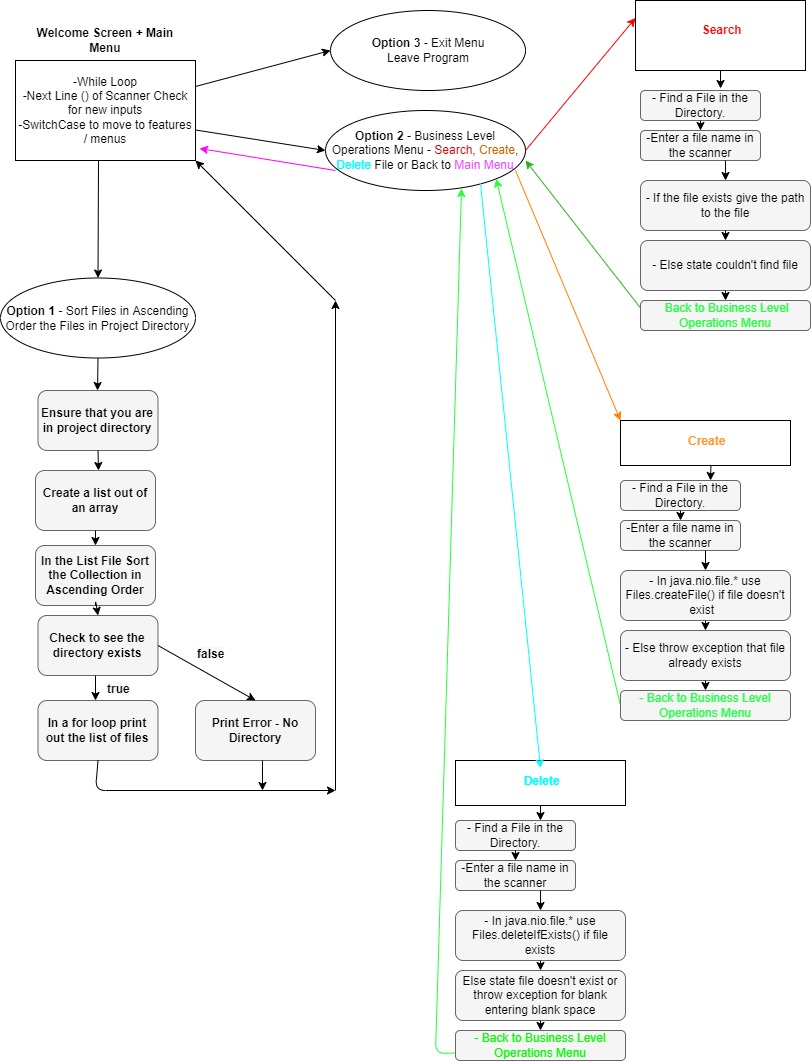
**PG FSD Implement OOPS using JAVA with Data Structures and Beyond – Project 1 – Flow Chart**



**PG FSD Implement OOPS using JAVA with Data Structures and Beyond – Project 1 – Sprint Planning**

**Sprint 1 – Welcome Screen – Menus**

1. Create a Welcome Screen that displays information about LockedMe Project
2. Create an input (scanner object) that paired with a switch case statement will allow for menu navigation. A while loop is likely required
3. Design the Main Menu test as a separate text to be called whenever needed to navigate to
4. Design a Business Level Operations Screen to behave like the Main Menu Screen – It requires 4 categories – Search, Create, Delete, Back to Main Menu
5. Design an exit option for the Main Menu

**Sprint 2 – Business Level Operation Functions**

1. Create the Search algorithm – Find requested input files
2. Create the Create new File algorithm – Create new files of your choosing
3. Create the Delete File algorithm – Delete the requested files of your choosing

**Sprint 3 – Create a File Directory Sorting Function**

1. The function must open a directory
2. Discover all the present files and place them in a list
3. Using collections.sort() the List must be sorted in ascending order

**Main Algorithms**

1. Create the Search algorithm – Find requested input files

* Request input from a scanner
* Create file object after discovering the file’s path
* If it exists print out the path of the file and state it exists
* If it doesn’t exist tell user the file isn’t found
* Back to Business Level Operations Menu

1. Create the Create new File algorithm – Create new files of your choosing

* Request input from a scanner
* Discover the file path and concatenate it with the file name and use that to create object
* Use try and catch statement in case there is an exception, and the file already exists.
* Back to Business Level Operations Menu

1. Create the Delete File algorithm

* Request input from a scanner
* Discover the file path and concatenate it with the file you will hopefully be able to delete
* If the requested item is found delete the file but otherwise throw exception for invalid inputs and when file not available
* Back to Business Level Operations Menu

1. Welcome Screen

* If the input is one of the three listed categories exercise the option using a switch case statement.
* The input comes from a scanner object.
* A while loop is used to keep the scanner running until you are requested to exit the application

1. Business Level Operations Screen

* Enter into the File Search, Create, Delete algorithms if you so choose using a switch case statement
* A scanner object is used to decide which option to exercise
* There’s an option to go to the main menu

1. Sort Algorithm:

* The function must open a directory
* Discover all the present files and place them in a list
* Using the collections.sort() method the list must be sorted in ascending order
* Use a for loop to print the list of ascending files if the directory exists
* Otherwise print “directory doesn’t exist message.
* Go Back to Main Menu

Ways to Improve the system:

1. The Folder of where all the files are created, sorted and deleted is hard coded. If I had more time, I would improve the system to allow a user to choose which folder or directory to work out of.
2. I used the java.util.collections.sort() method that I found online to sort the file items into an ascending order. This method was easy to implement. I did not find data on how efficient of a sorting algorithm it is. There may be better options out there.

Unique Selling Points:

1. It has a simple interface / menu system that allows users to easily navigate to the function the application can execute.
2. The menu screens allow for navigation to the previous screen and has messages to inform you if certain operation failed or succeeded. The user therefore is obtaining valuable feedback from the program when there are exceptions. All the navigating screens also give direction on what to do next.