Locked Me Application Simplilearn Specification

**Summary:**

This document is the specification document for the module 1 core Java programming course with Simplilearn. The project was to develop a prototype for an application that would allow a user to do file operations within a directory.

**Sprints:**

Number of sprints-2

Time period-2 weeks

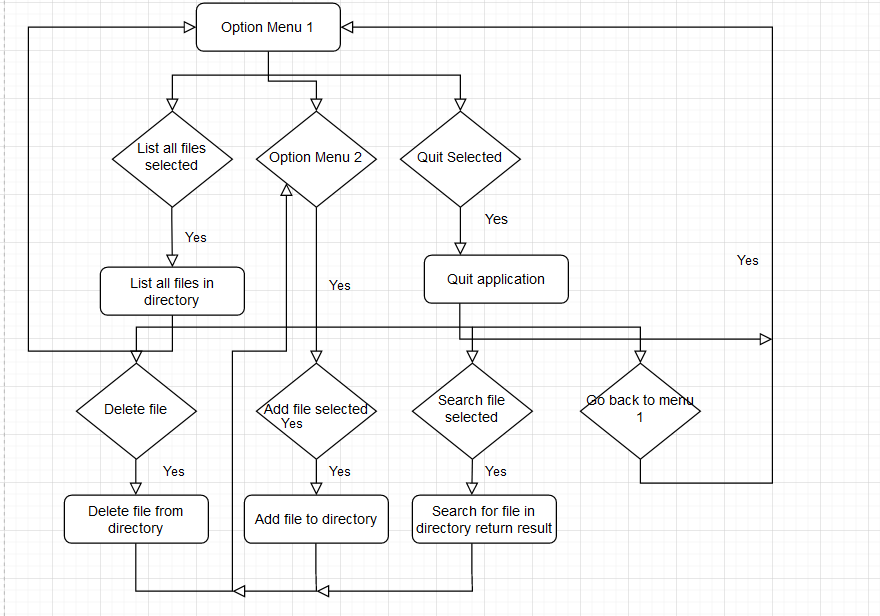
Sprint1:

* Plan: Think through code implementation and flow, class creation, class data, and then implement menu prompting.
* Deliverables: Java main function that prompts that displays the menu of options and prompts the user for input and takes them where they wish to proceed.
* Retro: I successfully delivered on everything I wanted to achieve this sprint but possibly could have spent more time on application clarity by logging more to the console to keep the user better informed. I also could have possibly done some ASCII art to make the console prototype stand out. These are both things I kept in mind going into the next sprint.

Sprint 2:

* Plan: Implement a file operations class that handles the file operations based on the given user input selection.
* Deliverables: A file operations Java class that handles all of the MVP actions outlined and is thoroughly tested.
* Retro: I was able to finish the application and delivered on everything I sought to deliver, but struggled with case sensitivity and was unable to include this functionality as a lot of the Java file functions handle case sensitivity on their own. Other than the case sensitivity I am happy with the prototypes state and my implementation of it.

**Flowchart:**

****

**Core Java Concepts Used:**

* Java project creation
* Git version control
* System logging
* Getting user input with scanner
* Encapsulation with class creation
* Private and public member variables usage
* Try/Catch blocks
* Java file creation, deletion, and writing
* Command line interface implementation
* Running and executing a Java project

**Link to Github Repo:**

<https://github.com/devonmcdonald00/LockedMeDotCom>

Conclusion:

To conclude, the application could be improved by adding a feature to change the private member directory variable in the file\_operations class that could be changed to somewhere on the user's local machine instead of being hard coded to the files directory in the project folder. Another change that could improve the application prototype is simply just including more file functionality such as truncating a file, writing to a file, printing the contents of a file, etc. The unique selling points of this prototype would include the user visibility through communication with the console, easy to use interface, and simple but effective backend functionality that makes this prototype stand out.