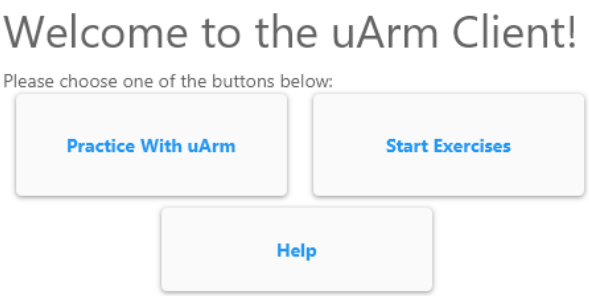
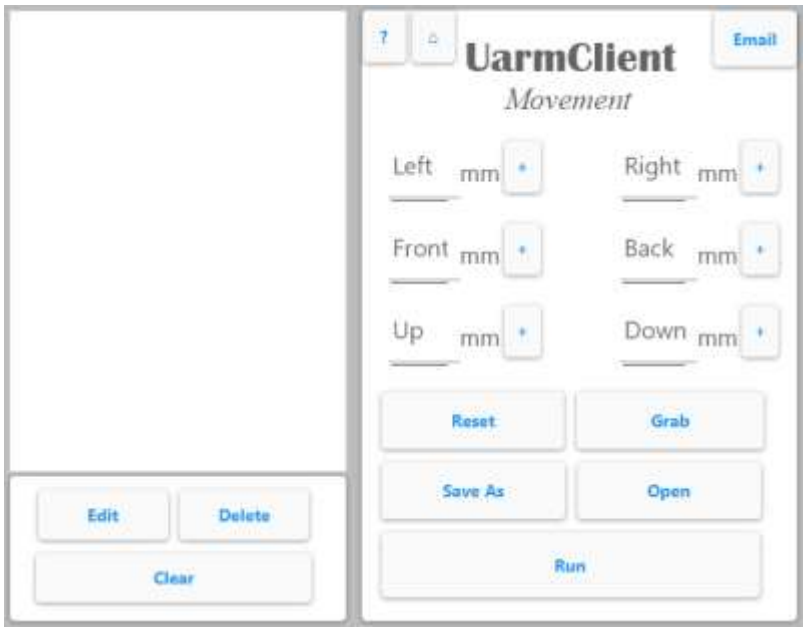
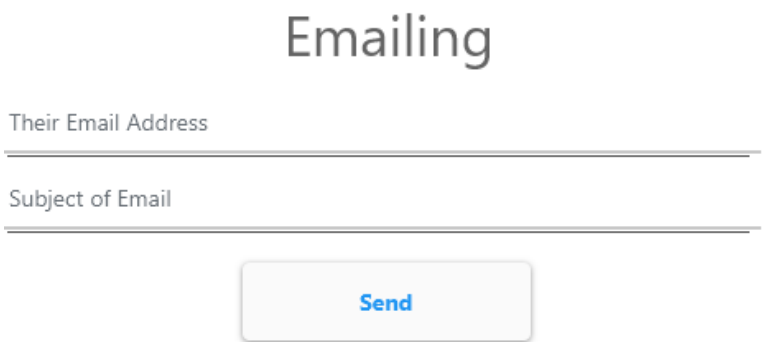
Criterion E: Evaluation of Project

Meeting the Success Criteria:

1. Create an aesthetic graphic user interface that can be easily utilized by middle school students.	**Met-** (see video) 
2. Establish a connection, whereby communicating with the robotic arm using the application on a computer using Java.	**Met-** Application can execute steps to make the arm move towards the targeted direction (see video between 0:21-0:35).
3. The application must allow the user to use an arm, by having a simple yet powerful graphical user interface.	**Met-** Multiple functions when using the arm (adding, saving, opening, editing, deleting etc.) (see video between 0:35-1:04). 
4. The speed of the arm should be slow so that it can be considered safe by the client.	**Met-** The arm's speed and limits are considered safe by the client (see video between 1:16-1:22 and Appendix 6).
5. The user can use all the functionalities of the arm: moving the tool head to a location and grabbing small objects.	**Met-** Activate and disactivate the pump, beep when a limit is hit and moves to desired location (see video between 2:34-3:36).

6. The application should allow simple management of a list of exercises that the students will complete.	**Met-** Multiple functions like sorting and editing for the exercises. (see video between 3:44-4:17) 
7. The application should allow the user to easily manage the files of the application.	**Met-** Application can externally and internally save and open files and functions without preexisting files. (see video between 4:28-5:06)
8. The application should allow the teacher to send exercises to his/her students.	**Met-** All the information of the application can be sent to others. (see video between 6:28-7:00) 
(See Appendix 6 to view the client's remarks on each of the success criterions as well as the recommendations stated below)	

Recommendations For Future Development

Minor Improvements:

Client:

- **Background:** information given to the user isn't effective (see Appendix 6).
- **Recommendation:** make most important words stand out.

Major Improvements:

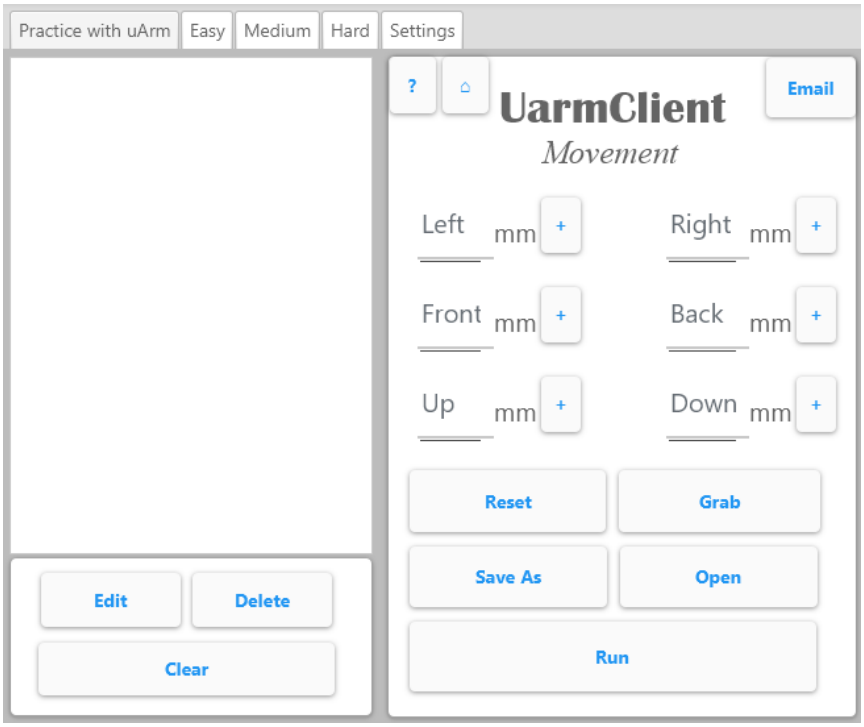
Personal:

- **Background:** As the uArm is mainly supported by Arduino and Python, not Java.[1] Initially, I used a python package to control the arm that was too complex. Then I used Arduino to change the firmware of the uArm, but that limited its functionalities (see Appendix 3). At last, after thoroughly understanding the protocols and advice from the supervisor, I installed a firmware and external libraries that best fitted my needs. This process took longer than needed (see Record of Tasks).
- **Recommendation:** First, understand all protocols of materials, then review all the methods that best fit the needs of the client. Start implementing the easiest one to decrease the unnecessary time of understanding complex concepts and techniques.

Client:

- **Background:** Application has too many windows (see Appendix 6).
- **Recommendation:** After designing the windows, evaluate them to improve the capabilities of the application. See Figure 1:

Figure 1- New Version (Design)



[1] http://download.ufactory.cc/docs/en/uArm-Metal-Developer-Guide.pdf

Extensibility

The client was satisfied with the overall aspects and functionality of the application (see Appendix 6). The client and I then discussed about the following modification:

- For future applications, I should make the code and GUIs simpler to improve the efficiency of the overall application (see Appendix 6).

  Personal Improvement to becoming a better computer science student**:**

- I should always thoroughly investigate the mechanics and protocols of the hardware beforehand since even though a connection was established, I wasn't able to comprehend the different configurations as I used trial and error to solve most of the problems I encountered.

Overall, this IA gave me valuable experience for future endeavors in the field of Computer Science.

Word Count: 484 (excluding Titles, Captions and Parentheses)