Ian Klein Ben Tucker Dave Geiss 4/24/15

CS347 Assignment 4

How to Run the SmartTimer9001:

Download our project from out github repository located at: <https://github.com/mobilehobo/SmartTimer9001.git> in the Iteration-Two branch. Make sure that you download the settings.txt file with the jar file. The only way our program works is if there is an already existing txt file called settings.txt. Once the project is downloaded, one can run the project by double clicking the jar file. Once this is done, a GUI window will pop up and it will ask the user to set the alarm. If the user clicks the set alarm button, it will bring up a new window and print out the current setting of the alarm to the console. The new window that is now open will have settings that the user can edit and also a browse option to pick a file path that could be used. Currently, the only files acceptable are executable files. Once the user is satisfied with their settings, they can click the save button. The save button will open a new window with the new settings and also save them to the text file called settings.txt. (Every time from now on when using the application the previously saved settings that are in settings.txt should load to the settings GUI). Once the time in the settings reaches the current time of your system, the respected reminder message or alarm message with the application chosen are displayed. To exit out of each window you just need to hit the x at the top. To close the program completely, hit the x at the top of the first window that was opened titled “SmartTimer9001”.

Our Unit Testing is very simple. We used JUnit to test all of the methods inside of our Alarm Class. We weren’t really sure how to unit test our Swing GUI class because there aren’t any methods that aren’t action-listeners. We figured that opening and closing the GUI was good enough unit testing.

To run the Unit Test just run the AlarmTest.java file. There should be no failures.

Also we were unable to figure out how to adapt cucumber and ruby to work with our Swing GUI but we tested each part of the GUI ourselves hundreds of times to make sure everything performed optimal.