Project 4 Report—Chrissy Soulakian and Cole Grover

During the last project, we had trouble with version control because Mercurial tried to create new branch heads when we would switch off pushing changes. For this project, we did most of the coding on Chrissy's computer and pushed changes to the repository from there. We both had the latest version of the project running on our respective computers, allowing us to test different ideas in identical environments. When we found a solution, it would be implemented on Chrissy's version.

With regard to testing, since we started with the Stopwatch project, we had to copy the Android Tests from ClickCounter into our new version of Stopwatch that would become project 4. From there, we edited the tests so they fit the requirements for project 4. We then went through the classes from ClickCounter to find the basic implementation and copied the classes over to our project 4.

As we were fixing the Time Model from the Stopwatch to count down, we realized that the functionality of the time model was very similar to the ClickCounter. We edited the time model to have both the functionality of counting up and down, which allowed us to omit the ClickCounter classes from our timer.  We also used the refactoring capability of Android Studio to rename the original states from Stopwatch to the new states we implemented.

We started the model with our team members without fully understanding the Stopwatch and Clickcounter programs. We came up with a diagram to the best of our ability and when we started to implement the model, we realized it would be easier to have an extra stopped state that only occurs when the timer has counted down to zero. As we continued to code, we updated our diagram to represent an effective implementation. We think it is most effective to start with a model and allow for changes as you code. We found that switching back and forth between coding and modeling was not only effective for this project, but also for us as a team. Because of our process, our final model is an accurate representation of our code implementation.