Progress Report Team 4
October 7 – October 20

Greg Becca
Dan Sheaffer
Andy Vanek

**Accomplishments This Iteration**

- Added grayscale filter
- Major refactoring within UI and
- Removed "TempImage" entirely from project
- Added tooltip history to Undo/Redo
- Added ShoppedGUI.cs that concentrates on event handlers
- Added more Doxygen comments

**Tasks for Iteration 4**

- Complete Undo/Redo
- Add more filters (e.g. sepia, saturation)
- Add more buttons on toolstrip for ease of use
- Create test cases (unit and integration)
- Start working on Crop (per professor's input)

**Problems This Iteration**

This iteration was a little rocky. We focused on refactoring and bug fixing throughout the majority of our time. We created more classes for a more organized, professional, and (most importantly) ease of programming. Also, some of our bugs were hard to squash. In one instance, we would use zoom, rotate, resize, and the outcome would be something radically different. This required the removal of TempImage and another issue in the bug was adding the functionality of first unzooming before tackling any other changes to the image. Adding Undo and Redo was implemented with a simple list. However, this simple list is full of bugs as shown in our presentation. Overcoming these pesky bugs has now given us room to concentrate on adding filters, test cases, and crop functionalities.

**Documentation Progress**

We've created our end-user manual and programmer's guide. The end-user manual will be attached to the e-mail with this document. Our programmer's guide is located on Andy's CS website. We decided to use Doxygen to generate useful, organized documentation. To do this you enter specially formatted comments in the code that will later be converted to HTML. The link is:

http://www.cs.kent.edu/~avanek/Capstone/ShoppedDocumentation

**Member Contribution**

We still believe that this project is not big enough to work on individually because of the lack of independent parts. If two people were working on this project at the same time there could either be code corruption or repeated effort. We are currently using the "pair programming" paradigm. Andy has been the main coder while Greg and Dan research and develop possible algorithms for our current focus. We have also found that having someone watch over the person typing leads to better, more efficient code, and less time spent debugging simple problems. During the next iteration, we should have an easier time distributing tasks because of testing. We also take turns doing documentation and progress reports.