Project 1 Retrospective Report

# The Good

Although this was the first project as a group, our teamwork was surprisingly good! Everyone contributed opinions and helped towards the completion of our project. Throughout the project, our group had some intense and heated moments. For example, we had a meeting where we argued about why some of the development work was necessary, such as the user account system. After having a team discussion, we all reached the same conclusion that this portion of the project added valuable functionality that went over-and-beyond the requirements of the project. Each team member responded to these meetings in a positive manner, using these moments to become a better member of the team. Overcoming adversity as a team and becoming a better unit after all is an important skill for any team. Our team exemplified this.

Another success that we had as a team was our communication. There was never a moment where communication lacked or fell through. Each team member was active in our online group chats and everyone participated in meetings. This made work on the project less stressful and helped overcome some of the issues we faced. One way we overcame an issue with communication happened at out last meeting. When we realized that we were going to be pressed, we decided that we needed to reallocate work. One of the team members began writing the final report, while the other three members finished up development work on the project. Each teammate talked about their strengths so we could best distribute the work so it would get done.

During development, each team member worked hard to finish the work they received. Everyone stepped up and contributed to get work done when we were in crunch time. Also, each teammate was able to use their strengths to benefit the team. For example, two of our team members were strong with node.js and another teammate was strong in AngularJS. This allowed team members to take time and learn the technologies that they were not familiar with, as well as the ability to feel comfortable during the project.

# The Bad

Through the course of the project, we ran into several issues with the technologies that we planned on using. The first issue we discovered was with Angular 2. Our plan was to use Angular 2 on the front end to help us with front end controllers, services, and routes. We were able to install Angular 2 and start using it to make a simple application. When we attempted to add different routes to the website, that’s when it all went wrong. The current routing package in Angular 2 was not compatible with the Angular 2 version we were using. When we updated the Angular 2 version, we were then unable to load any webpages. After doing a few hours of investigation, one of the Angular 2 Javascript files had a 404 error, causing the application to fail when loading. To alleviate this issue, we resorted to using Angular 1. Angular 1 allowed for the same functionality and was a tool that each teammate was familiar with. Although we were not able to accomplish or secondary goal of learning a new technology, Angular 1 was a perfect solution for resolving the Angular 2 fallout.

We also had issues with setting up OpenCV. At first glance, we thought that using OpenCV would be as simple as running a quick package manager install command, including the package in the source code, and writing a few lines of code. This turned out to be not true. OpenCV was very difficult to install. The first issue is that the open source library we planned on using was only compatible with OpenCV version of 2.4.11 or less. The current version of OpenCV is 2.4.13, so we needed to search for an older version. We were able to locate an older version of OpenCV (version 2.4.11) and used that instead. To ‘install’ OpenCV means that you have to compile the actual source code. In order to compile the source code, it was required to install Python, CMake, and Visual Studio. After getting these programs installed, we were able to get OpenCV working. The issues with installing OpenCV held up development work on the facial recognition portion of the application for several days. One of our teammates was able to get OpenCV working and he championed this side of the project. The rest of us focused our time to other portions of the project, such as the class management system.

We also faced issues outside of the technologies that we used. The first issue was that we started the project way too late. After getting the project assigned, we did not utilize the first week to our advantage. After getting the project proposal completed, we did not get development work started until after the following weekend. If we would have started the project as soon as the proposal was completed, we would have had less headaches at the end of the project. We also had issues with over committing ourselves to the project. As a team, we thought we would have more time for the project. As it turns out, we did not have as much time as we thought. Our focus should have been more on the Minimum Viable Product, rather than adding features on top of that. We added an account system and class management system on top of the face recognition portion of the project. This caused development resources to be allocated in places other than the facial recognition part of the project.

# Action Items

For the next project, we are going to use our successes and build on them and try to fix our errors. One of the goals for the next project is to keep up our good communication and teamwork. If these two aspects of our team stay constant, we should have no issues. Our next goal is to start the project early. If we start the project early, we will be able to produce a better product at the end of the project. It will also make our lives a lot less stressful. Another goal is to do better initial research on the technologies we plan on using. Both Angular 2 and OpenCV turned out to be more difficult than we thought. We wasted quite a bit of time trying to get these technologies to work. For the next project, we will make sure to better research the technologies we are going to use and make a more informed decision. This will hopefully lead to faster development times and a better end product. Finally, we are going to focus more on the Minimum Viable Product first and then adding fancy features. Hopefully this line of thinking will result in a better grade. After all is set and done, project one was a great learning experience and we hope to carry all of this knowledge into project 2!

# Summary

|  |  |  |
| --- | --- | --- |
| Deliverable | Delivered/Missing | Comment |
| Take a picture and separate faces | Delivered | To accomplish this deliverable, we used OpenCV and Kairos. To upload the picture, we used an open source project called webcamjs. |
| Compare the faces against the pictures which the instructor uploaded via the manage class screen | TBD | This is accomplished using Kairos again. An image is uploaded to the database and stored with student information. |
| The program will run via the instructor’s computer | Delivered | To accomplish this deliverable, we made a MEAN stack web application. This includes MongoDB, Express, Angular, and Node.js. |
| Unidentified People will be asked to be saved to the database for future reference | TBD | TBD |
| Manage Class system | Delivered | This was a stretch goal that we wanted to accomplish for the project. We successfully completed this deliverable. We used the MEAN stack to display and manage instructor’s classes. |