# Professional

## Communication

Our groups communication was decent, and we were in contact nearly every day about where we were at with our currently assigned task. We used a Facebook group chat to organise meetings, scrums, and to find time when we were all available to do some work together.

Due to each person having quite different schedules and commitments we were only able to scrum with the full group approximately three times a week, but would often scrum with whoever was on campus on the days where the full group couldn’t be in. The times where we could have a full scrum were the times where we had the best picture of where each team member was at with their progress on the current sprint. Being able to scrum every single day would have greatly helped as we would have known if someone was struggling more than the others and could offer some in person help rather than help over a messaging app. The Agile principles say that face to face communication is better than the alternatives and our group certainly found this to hold true, especially as the project got bigger and more complex.

Using a Facebook messenger group chat was perfect for making up for lost scrums due to our group’s day to day schedules. We all had Facebook messenger already installed on our phones and our we had our phones on us nearly 24/7. Any time of the day if somebody was stuck on a piece of code or didn’t know how to do something in Laravel they could message the group and they were guaranteed to receive a reply promptly.

One of the few times of the week where everyone was free was Thursday up until class at three. We made up for lost scrum time by sitting down together at this time and getting everyone’s assigned tasks up to date, helping those that needed help, and pushing any complete features out to the live branch. Because of this I felt that Thursdays were a valuable day for us and it was the best alternative we had to a lack of consistent scrums.

## Leadership

Early in the semester it became clear that a couple of the group members needed to take a leadership role as two to of the others weren’t focused and didn’t seem to have as clear of a picture about what needed to be done. For the most part Corey took the lead in terms of setting up each sprint, and if Corey was busy Josh or I would step in and set it up. Delegation of sprint tasks wasn’t decided by any one person; everybody had a say but final confirmation of who was doing what mostly came down to Corey and myself. A vast majority of this was organised in class on Corey’s account as he’d take the initiative to do most of the driving.

The times where I took on a leadership role was on our Thursday group work sessions. Most Thursdays I would talk Callam and Prasanth through the logic of whatever task they were working on as I felt the most proficient with programming logic. I would often have my task finished in one sitting which would free me up to help or guide my team mates.

## team work

Our group worked well as a team well. Two members didn’t contribute as much to the work as the others due to not being as proficient technically but those that were technically proficient worked well together to overcome some of the surprise challenges we faced.

When Laravel was introduced it really through the entire team off because it was unlike anything any of us had dealt with before. Corey, Josh and I were all working on the adding to the database the semesters, courses, and assignments respectively. We really struggled with how it all worked in Laravel regarding how views and databases were handled. Eventually I figured it out and pushed my code, at which point I was able to work through it with Corey and Josh as they were coming up against all the same errors I had just worked through.

Thursdays were where most of our core team work was happening. We got a lot of work completed, especially if we could help finish up the tasks of the less competent members. One Thursday in particular we had two members not show up or let us know what was going on with their tasks, so Josh and I split up their work and with Corey’s help worked through the tasks to get them finished and push to the live branch before the sprint deadline.

## time management

Certain members of the group managed their time very well. I personally don’t feel like I managed my time very well. This has been an issue I’ve always had and need to work on. Early on in the semester my time management was a lot better than towards the end. During the final sprint I had a lot going on in my personal life and didn’t contribute to the group very much. I feel like I let my team down right at the end.

One of the ways I was able to maximise my time was finding bits of code people would commonly use and adjusting it for my purposes. This saved me from figuring out how to code something that was readily available online. This was particularly helpful with JavaScript and CSS.

## motivating self and others

At times it didn’t feel like some members of the group were particularly motivated, especially when we were having to deal with tasks that we were really struggling with. The biggest motivating factor was that there were four other people relying on you. The days when I really didn’t want to have to deal with the project, I thought about how that would affect my group mates, and that gave me the push to get on with it.

It was easy to motivate some of the group members more than others. Two of the group members were very work focused and the other two weren’t. I attempted to speak with the less motivated team members about how we were relying on them to get their tasks done and when this didn’t seem to work, I let them know I was disappointed and that they were letting the rest of us down.

## Agile

Throughout the course of this project we aimed to adhere to the Agile Principles as much as we could. Some of the principles we consistently met, and others fell a little short. Each sprint beyond the first aimed to have a new feature integrated and deployed into the software. If the client had new ideas that they wanted added to the software our team would create a new ticket and start work on that issue in a sprint in the near future.

Once we had a working foundation our team was able to make changes to the software easily while still maintaining a usable and working tool. When speaking with the client we would take note of what they wanted added or changed and distilled this down into several user stories. This broke larger tasks down into manageable sizes for each team member to work on.

An example of this within our project was the need to add a Laravel framework. Once the framework was up and running, we were able to take what we had already worked on and integrated it into the framework. There have been several user stories where the implementation wasn’t quite how the client had envisioned the feature working and there was a need to revise that feature in a future sprint.

Agile workflow is a constant process of working on a feature, implementing the feature into useable software, reflecting on the sprint, and beginning the next sprint. In the big picture I feel like we were successful at achieving this, although every team member, myself included, did fall short of implementing a complete feature by the end of a sprint at least once.

# Technical

## New Technology

Over the course of the semester there were several new technologies introduced that I had used either very little of or had never used. These included JavaScript, Git, and Laravel.

Early in the project I had to code a countdown to a date that was specified by the user. This was ticket #16: Days left to complete Exam. Prior to this I had never coded anything in JavaScript. I used w3schools to get up to speed with JavaScript. I used their count down app tutorial as a basis to work off. The biggest challenge here was figuring out how to take user input and turn that into a Date variable that was usable in the code. Once I had the day the user wanted, I discovered that I could add a one to four to the start of the semester date to get to the day of the week the user wanted and then add a multiple of seven to get to the week of the semester that was desired.

At the beginning of the project I had only used Git a very small amount. Learning the Git workflow was not too difficult although there were a few issues with how our team used Git early on. Knowing how to deal with merge conflicts early on was something I and the rest of my team found difficult. Our first pull request was a disaster, and we ended up copying code into master rather than merging with master. Once that sprint was over, I spoke to the team and we all agreed that our branching and merging was a mess. To overcome this, I made sure that I was involved with all pull requests in some capacity to ensure that all merges went smoothly, and that no one took any strange shortcuts, such as what had happened previously. Another aspect of this was making sure branching was being used effectively. Due to the group not knowing how to properly do a pull request early on, we had a few dead-end branches. Going forward after that I made sure that branches merged back into master once the branch feature was functional.

During Sprint 3 the Laravel framework was introduced in ticket #23: Laravel Framework. This was the largest challenge that I faced during the entire semester. The entirety of this sprint was getting the group up to speed with how Laravel works and how it integrates with the work we had completed already.

The largest roadblock I had with Laravel was implementing the ability to add an assignment to the database and display all assignments to the user. This was ticket #26: Store Assignment Dates & ticket #10: Due Dates. I had to go back through Laravel tutorials and read documentation to figure it out. I came up against a dozen errors and made note of how to fix each one because I knew Corey and Josh were working on very similar tasks at the same time and they could use the help.

The process of getting this set up in Laravel first involved creating a new model and migration from the command line. The second step was setting up the naming of each of the database fields correctly. After the database was creating correctly, I had to set up the routing so that the buttons would lead to the right places. Once that was done all that was left was to copy the body of the tutorial code and rename all appropriate fields so that they would work with our database. I feel that we struggled with this more than some of the other groups. Despite the struggle, this is the thing I am most proud of accomplishing because it felt like a huge challenge for me personally and being able to help my team mates through very similar tasks was extremely rewarding.

## Design

The earliest ticket I was assigned was #4: Team Information. This involved writing up bios for each team member so that the user had some sort of understanding of those that made the tool. I asked my team mates what they thought I should write about them, but I didn’t get any helpful feedback. I designed a basic team info page with cat profile pictures to go with our team’s cat theme. This involved some minimal inline CSS as we did not have a finalised CSS file at that point. My team mates as well as the client expressed that they liked the approach that I took with the info page.

During the end of one of our sprints we received some feedback from the client that there was too much scrolling in our site and that it should be minimised. As a result of this I was assigned ticket #20: Compress FAQ Page. I decided to make each FAQ topic collapsible as this would save a lot of screen space. I followed along with a w3schools tutorial on making elements collapsible. This involved a combination of HTML buttons, some CSS, and a small portion of JavaScript.

## Systems Administration

Sprint 4 involved getting our site running on a server. This was assigned in ticket #29: Running on a server. To initially get it on a server several of the group members worked together so that we all understood how it all works.

The first thing that was setting up the database on mariadb and then cloning our repository onto the live server. As a group we struggled with this initially as we collectively misunderstood several of the steps. It wasn’t until the following week that a better grasp of it all fell into place. After that initial sprint Corey became the teams ‘Systems Admin’ and handled deploying the latest version of the software.

## Version Control

From the very beginning of the semester our project has been version controlled using Git. I am very confident in my ability to use Git now compared to the start of the project. Initially I had to make sure I was adding to be committed only the files I had been working on, as prior to this course I had used “\*” to add everything to a commit.

The feature that made the biggest impact on the way I used Git was branching. Anytime I was working on a new feature I would start a new branch and work there until I was finished and ready to merge back to master. The first pull requests that my team and I dealt with did not go well, and I kept a closer eye on all pull requests following to make sure everything is being used correctly.

There were several moments where a team member or myself had made some unfortune changes that they wanted to discard and roll back to before they began working. This led to learning about resetting the head on a branch.