**Lessons Learned**

Over the course of this term, we used a lot of different methods for developing both C# desktop applications and ASP.NET web solutions. In the end, the ones we decided to go with were MVC with entity framework for ASP.NET and using LINQ to SQL as the framework for C# desktop apps.

Our team had a few issues with scheduling due to family and health issues, but we were good about dynamically adjusting workloads and coordinating online to keep everyone informed of who was doing what and who needed things taken off their plate. One thing we didn’t do so well was analyzing the requirements before splitting up the work at the beginning. So we’ve learned that in the future we should think in a more detailed way about what each requirement actually entails and how much work it would take, so we can make a better division of labour up front.

**Workload Distribution**

The work on the project was split up dynamically in response to the situations that came up for each of us. During Workshop 4, Angela’s family health issues were at a peak so she had the lightest piece, whereas the portion that Wade picked turned out to be the most complex piece and Tom’s was in the middle.

* Products & Login front end: Angela
* Packages and login backend: Wade
* Suppliers/Products\_Suppliers: Tom

Going into Workshop 5, Tom was having a long illness and ended up with the lightest task at first. Wade came out of the gate sprinting, but his family emergency took the forefront toward the end. Tom picked up the slack as he recovered so we could give Wade space for that. In this phase Angela ended up with the hardest piece of the core requirements and spent the most out of class time of all of us.

* Task 1, any associated login functionality, and theming: Wade
* Task 2, contacts, recreating Workshop 4 within Workshop 5, and misc. cleanup: Tom
* Task 3, booking system, and home page: Angela