# Content of 200 and 300 Level Courses

During the course of my study at CPIT I have accumulated knowledge and a number of skills. This knowledge and these skills have been vital in the successful completion of the project. In this section you will find the courses I have completed and how the knowledge and skills acquired from said course were useful during the project. Where it is applicable I have noted knowledge and skills that were not explicitly taught and how they may fit into the curriculum so as to enrich the courses CPIT has on offer.

## BCIS201 Alternative Modelling

### Diagrams

During this course the importance of modelling a system was presented. It allows for a simplified version of the product and a focal point for discussion among interested parties. During the project we modelled the system architecture and used class diagrams to model the application’s internal classes. We were then able to discuss issues during development with each other and other interested parties.

### Joint Application Design (J.A.D.)

We also learnt about J.A.D. in Alternative Modelling. The concept of including the client who takes an active role during the development phase is a basic yet powerful one. They are after going to be the end user or have a close relationship with the end user of the product. It makes sense to utilise the knowledge they possess.

During the project a McKesson employee acted in that role for us. He attended our weekly planning meetings and end of Sprint reviews as well as some of our daily stand ups. It was he who would be using the product once it had been completed. When time was short his input as to where the effort should be focused was invaluable.

I believe having the client involved increased the team’s motivation and moral. It was clear that we were developing a product that was valued and would improve someone’s workflow.

### Scrum

As an alternative to the ‘Waterfall Model’ Scrum was discussed during Alternative Modelling. Having foreknowledge of Sprints, backlogs, and Scrum teams among other aspects of Scrum, allowed us to implement and work effectively within this framework during development.

## BCIS202 Systems Design and Implementation

### Project Management

During this course we examined the importance of project management. Although Gantt charts and critical path analysis have their place we didn’t feel the need to use either. During the project we managed the workload using the product backlog and prioritised tasks while consulting with our client.

We worked through the phases of the project, Initiating, Planning, Executing and Closing Down. We also ensured our project was successful by meeting the success factors.

### Risk Management

We put what we had learnt about risk management into practices. We identified risks. Analysed those risks. We devised strategies to enact in case the risk was to eventuate. Each of us individually monitored our own risks as we saw fit.

The course may benefit in the future from adding a practical application of Risk Management theory. Actually seeing a risk management plan in action, and the types of threats to a project one needs to manage would be of great benefit.

### Usability

We applied what we had learnt about usability in the web browser user interface. We used conventional controls in a conventional way to make it easy for the user to configure the application for their own needs.

## BCPR203 Database Management Systems

### SQL

During the project we had to read and manipulate data held in Microsoft’s SQL Server 2012 Databases. We had learnt about databases using MySQL and the transition to Microsoft’s version wasn’t very large. We were able to use the knowledge gained from the course and a little bit of reading to learn the required variant and write appropriate queries.

### Relational Database

Having learnt about Relational Databases were able to write complex queries to get the information we required.

### Data Warehouse

We also had to extract information from the Data Warehouse and ensure it was running. We had been presented high level information regarding this feature but perhaps the course could have benefited from a practical demonstration or exercise to develop a deeper working knowledge of this feature of SQL databases.

## BCIS206 Professional Practise

### Communication

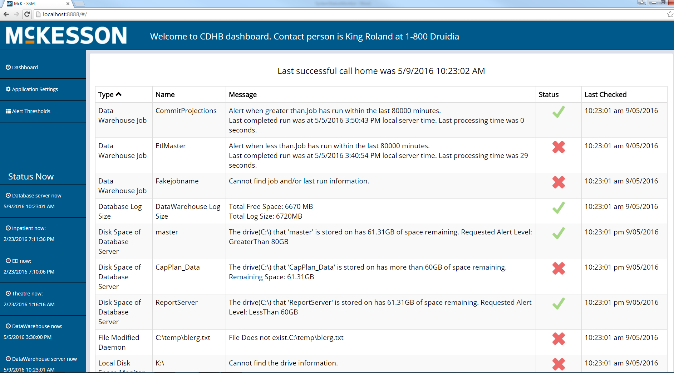
During this course there were many opportunities to practice effective communication. The communication skills practiced during the Professional Practice project, with partners and the client was directly relatable to communicating with the individuals involved in the BCCE301 Project.

### Managing A Project

The practical application of project management during the course has aided with the day to day management of the project.

## BCIT242 Website Development

### Html

During this course we were taught a basic introduction to website development using HTML5 compliant code. This knowledge aided in the development of our User Interface. We used tables, buttons, and input fields to create an easily navigable UI.

### Responsiveness

We were also introduced to the idea of designing a responsive webpage. During the project we created a webpage UI that was somewhat responsive. As the webpage reduces in width the content adjusts so that it remains viewable.

## BCIT252 Multimedia and Animation Development

### Media Elements

While the project didn’t call for much in the way of multimedia or animated elements we did uses images saved in formats that were web appropriate. This ensures the UI can be viewed with major web browsers such as Internet Explorer and Chrome.

## BCPR280 Software Engineering 2

### JavaScript

During this course we learnt pure JavaScript to code. This knowledge was invaluable when creating the client side user interface. We were able to write functions that executed when required, such as sort by column when the column header is clicked.

### Angular

To help with the manipulation of dom elements in the project we used a JavaScript library called Angular. This made it easier to collect and manage the user’s data when changing settings. This may have been of benefit when working on our Software Engineering project along with learning about dependency injection to manage modules and functions.

### Agile vs PSP

During this course we were tortured with Personal Software Process(PSP). We tracked errors and the time it took to code functions in an effort to increase our ability to estimate how long it would take to code something. Also by partaking in this time devouring exercise we were ‘improving our coding’. I found this endeavour fruitless and suggest an alternative to improve the quality of code.

I believe it would have been better to code small simple projects each week and then evaluate this at the end. This could be done by peer review. Each pair could evaluate each other’s code and make suggestions about how it could be improved. Then the second week the same task could be undertaken and the code refactored. This way students will improve the quality of their code by learning better ways to do things, rather than creating data that ultimately had little to no use.

## BCPR282 Best Programming Practices in Java

### Separation of Concerns

We learnt about keeping code in loosely joined classes that do one thing well. In our project we ensured classes had a specific role.

### Dependency Injection

Classes were loosely joined and Interfaces were used, e.g. in the creation of the daemons. The daemons were classes that set about collecting the required data when they were told to run.

## BCPR283 Best Programming Practices in .NET

### Unit Tests

During this course we learnt how to write unit tests to ensure the quality of the code that was written. We had used Visual Studio, the same IDE we used during the project. This made it easy to create a unit test suite and write tests for our code.

One thing we could be taught during this session is how to open up classes for testing. It was difficult to test certain aspects of our project with unit test due to the encapsulation settings. An explanation and experience with using the internal declaration could be helpful to students in the future who would like to open a class up for testing.

Also demonstrating how interfaces could be used to create “mock” classes that can behave exactly how you want them to when testing a class that has a dependency on one or more classes.

## BCIS285 Software Applications and IS Testing

### Higher Level Testing

This course was predominately a high level approach to testing. We looked a test management, test estimation and test monitoring. As a programmer I would have liked a lower level approach. This however did give me a good understanding of the importance of testing. Improving the quality of the product through competent testing is an ideal one should strive for.

### Test Driven Development

Perhaps looking a test driven development could enhance this course. During the project we worked in a feature driven style. Testing was done once code was written to ensure it performed as expected. I wonder if more time spent planning the classes and how they actioned their duties may have led to better code.

## BCPR294 Server Side Web Programming

### SOLID

During this course we learnt about the SOLID principles. These principles were helpful when designing our project. We ensured classes obeyed the Single Responsibility principle and only managed one thing.

### Feature Driven Development / Cucumber

We learnt to code in a feature driven development style. Cucumber was used to help define those features. When working on the project we also worked in a Feature Driven Development style. However, those features were not strictly defined unless they were complicated. Using cucumber here could have helped, especially when some features were ambiguous.

## BCPR301 Advanced Programming

### Design Pattern

I think design patterns have been the best thing to have learnt about during my study for my degree. I like how the generic nature of the patterns ensures they can be utilised in many different circumstances. I think using these in the earlier programming courses would have been a great way to improve code quality.

During the project we made use of the factory pattern to load the needed classes at run time. We had ‘daemons’ of different types that would be instantiated if they were required.

### Bad Smells

An awareness of bad smells in the code helped us to consider alternatives to some code. Often I refactored my own code to ensure it was readable thus ensuring those who work with my code in the future would understand its purpose.

## BCIS301 Management of Information and Communication Technologies

### Project Proposal and Planning

Creating a proposal and planning a project set us up well for our current project. Although a lot of the ground work had been done for us as McKesson knew what they wanted. We were able to understand how our product would fit into the current system and how it would enrich the working experience of the front line staff the would use it.

We were also able to plan an acceptable project that met the requirements of both McKesson and CPIT.