# Week 11 (3/Apr/15)

Blog<https://amulligant00122209.wordpress.com/>

Monday was mainly spent working with two different WebSharper concepts, formlets and flowlets. A formlet is the equivalent of creating an html form with JavaScript error checking. A WebSharper formlet has built in functions to check for empty fields, check if a field’s value is in correct format and disable buttons while fields are invalid ect. Each of these functions can be done with one line of F# code. So a complex form can be written in as little as 10 lines. A flowlet is the idea of joining n number of formlets together, these can be easily joined by using the <\*> operator. This process make is extremely easy to create complex forms like you would see in the likes of signing up for Amazon or any retail services where one form leads to another and so on.

Tuesday we had dev training, this week the topic was monoids. This is an aspect of programming which I had never even heard of before the dev training began. It was a concept I found really hard to grasp. Monoids are a notion which are used extensively throughout functional programming without ever even knowing they exist. Form my limited understanding of monoids so far they are patterns which a mathematician look for in computations. These patterns have to follow a set of specific rules to become a monoid. This is very rarely used but maybe something if you were able to fully understand could be beneficial in the future.

Wednesday I was asked to write a document that covered how to set up and integrate all three aspects of the current project I am working on. This document is to be kept for feature reference, and make the process of setting up PostgreSQL with an F# MVC Rest web API as well as setting up WebSharper

Thursday all the developers had a workshop on F#. Some of the more junior developers had recently been on an F# course in Dublin. They had put together a power point presentation to demonstrate what they had learned at the conference. This covered everything from the simple basics all the way up to using F# for domain drive development.

Friday was a relatively quiet day, I had a look at deploying a WebSharper application, and this was a relative easy process. The hosting I choose to try out deploying my application was AppHarbour. This hosting allowed you to import your solution via GitHub and simply click a button and build your solution. AppHarbour then gives a visual indication of if your application has built properly. If it is successful in building you can navigate to your application via a provided URL. I also had a meeting with the lead software engineers in regards me staying on at Continuum Commerce Solutions beyond the finish of my work experience. I will have a think about this over the long weekend and give them an answer next week.

# Week 10 (27/Mar/15)

Blog<https://amulligant00122209.wordpress.com/>

I was out on Monday

Tuesday I began having a proper look at WebSharper. On a first look at the documentation for WebSharper it appeared that there was plenty of examples to follow. But once I started to try implant some of these examples I quickly ran into problems. WebSharper runs clients side code as F# but you can annotate a function with [<JavaScript>], this means at compile time the F# code is translated to JavaScript and render within a div in the index HTML page. At the beginning I couldn’t figure out why the JavaScript annotated function would not appear on the html. I tried all the usual sources like Stack Overflow and Code Project for an answer but there are very little people using WebSharper.

By Wednesday I was getting more and more frustrated with being unable to even complete the simplest of templates with WebSharper. I decided to contact the creators of WebSharper itself. I set up a Twitter account and contacted Loïc Denuzière and Adam Granicz the latter tweeted back with some helpful books which luckily we had in the office. I like to use videos to learn but there is very little out there on this medium. After reading through some of the material I discovered that Visual Studio is required to have the latest updates in my case (Update 4) and the latest version of Nuget Package Manager. After waiting and getting I.T.S to install these on my machine I was finally able to see WebSharper actually work properly.

Thursday I began working with WebSharper properly, adding css styling to the F# code. This is add to the main HTML page as you would normally “<link rel="stylesheet" type="text/css" href="StyleSheet.css" />” and a class or id is added to a div or any other element with “Div [Attr.Class "class1"] -< [Any other code]”. I then began to add simple button handlers to the client code which called server side code and return a simple response like “Hello World”. As soon as I had this working I began to create more complex examples with form’s and sending multiple pieces of data across to the server either individually or a tuple.

On Friday I started setting up my project so I could use a WebSharper form to request data from my F# API via the WebSharper server side code. So on the WebSharper server side I dynamically create a HTTP request which gets a reply from the API and then serves it to the F# client side code before being rendered to the screen. An example of this is shown in the blog,

# Week 9 (20/Mar/15)

Blog<https://amulligant00122209.wordpress.com/>

I began Monday by having a quick look at the project diagram I had been given, and by taking a closer look at the technologies which would be involved. I have added the diagram to the blog and will give a more detailed description when I fully understand the work involved. I began by looking at types of APIs services like SOAP and REST. As it is an F# service which is required REST was the best fit as Visual studio now provides a downloadable template to run an MVC (model view controller) F# template. This template is relatively easy to use, follow and to create different APIs. Not much in the line of F# example exist but there is endless C# example which are straightforwardly translated.

On Wednesday I managed to create my first HTTP request which is a simple get request which calls into my F# API. The API then returned a simple 'hello' as a response. I then began to build on this by getting the API to return different formats like json and xml. The next logical step was to try send more complex requests like put and push where information is send in the body of the request and access this data through records the F# equivalent of a class(record.property)

On Thursday I began to advance the project a little further by add further functionality to the API by allowing the API access a PostgreSQL database and perform a query or insert data depending on the type of information sent, then return the result of the request back to the user. This set up was fairly quick as I had most of the code previously from my invoicing application. This concluded a mwp (minimal working product) and I will begin looking a WebSharper tomorrow. Also on Thursday I gave dev training for about 45 on PostgreSQL. My presentation last Friday was mostly theory based, but I mentioned having written queries using PostgreSQL concepts like json, jsonb, custom types, arrays and inheritance. I talked the development team and the database administrator through the queries and how they worked as well as different ways to access the data and the efficiency of different ways of performing queries.

On Friday I began looking at an F# web framework called WebSharper. WebSharper allows the developer to work with JavaScript and F# to create web pages. On first look WebSharper appeared to be a technology that would be easily enough picked up but after looking at it for most of Friday it going to take another bit before getting head around it. It’s not a very popular technology and there is a lack of documentation and tutorials. I have set up a twitter account and tweeted the creators of the language, I am awaiting their response.

# Week 8 (13/Mar/15)

Blog<https://amulligant00122209.wordpress.com/>

Monday I had a look into replication in PostgreSQL and the different ways PostgreSQL deals with scalability, replication and high availability. PostgreSQL has three different varieties of replication and each of these can be built up on to create very large distributed systems.

On Tuesday I set up two Ubuntu VM to try and replicate replication with PostgreSQL. I started by setting up the easiest of the replication protocols which is called “warm standby”. A server which is in warm standby mode is in continuous recovery mode and also known as a slave server, the other server then acts as the master and is in constant archive mode, so ever transaction written to the master gets written to a WAL file (Write-Ahead Log). These log files are then sent to the slave server periodically. If for any reason the master may failover, the log files can be replayed to restore the database with a minimum loss.

Wednesday was my final day of research into PostgreSQL, it was the case of checking the information that I had and adding a few final details.

Thursday was spent mainly working on the power point presentation. I have added the power point to the blog as I didn’t really write any code at all this week.

Friday was the day of my presentation on PostgreSQL, I spent the morning preparing for the presentation, by adding a few diagrams to the white board which I was going to use and making sure my power would work on the company laptops so it could be used with the projector. The presentation began at 10.30. The whole development team attended the presentation. I felt a lot more comfortable this time around. The presentation went very well I think and they want me to give dev (development) training on PostgreSQL this Thursday on PostgreSQL basic and run through the querying language and GUI environment rather than the mostly theory based material covered in the presentation.

I have already been given my next project which will initially be a simple F# application using WebSharper an F# web framework which uses a rest API to access a PostgreSQL database. This will take the next few weeks, and will provide more detail next week.

# Week 7 (6/Mar/15)

Blog<https://amulligant00122209.wordpress.com/>

Monday I meet with David and Thomas to talk about how my presentation and general work had gone over the first six weeks. They informed me that everything had gone swimmingly so far and the hierarchy where please with proceedings. They also informed me of my next project which I had expected to take up the next six weeks but I have been given what they call a “spike” or in layman’s terms an investigation of PostgreSQL. I had previously used this in my prior project but hadn’t went into it in any great detail. I have been given a few header under which the company would like to know in more details how things work. Also a new version of PostgreSQL has just been released with some great improvements in the storing of json.

On Tuesday I began my investigation by looking into who created PostgreSQL, where the initial idea came from. PostgreSQL prides itself on being open source. It has a huge community of people behind it which are funded in part by companies like google, Spotify and Skype to name a few. One of the unique selling points of PostgreSQL is that hundreds of extensions can be added to modify the initial product and morph it into something not even some of the most expensive of products like Oracle can match. PostgreSQL stumps the rest with its endless list of types it can hold within a column. PostgreSQL can hold an array in a single field or even a json object of up to 1 GB.

Wednesday I started to play around with the PostgreSQL query language. I began by finding simple tutorials and following them. Much of it syntax is similar to what I have been used of with Oracle and MySQL, but where it differs considerable is when dealing with json. Since the release of PostgreSQL 9.4 there has been considerably advance in json storage. Json can now be stores in PostgreSQL in a format called jsonb. This enables the uses of indexes on json columns. It is a bit strange do when comparing the size of json and jsonb fields, for some json columns json can be larger whereas on other columns jsonb can be larger.

Thursday I began to look into the different types of indexes used with jsonb. A GIN index is used with jsonb columns. GIN indexes are used when large amounts of duplicate data occur. In a GIN index is like a btree which has been heavily optimised for performance. Within a GIN index any duplicates will be added to what called a page list, if this page list fills up a paging tree will be created. If a lot of delete or updates are performed a Gin index will quickly become fragmented. But as I am performing it on a json column which are designed not to be changed this shouldn’t be a problem for json.

Friday was spent looking into scalability, maintenance and high availability. I started by looking at replication. This is where one server is set up as a master with n number of servers acting as slaves. The data which is written to the server is replicated to the slaves, if the master for some reason goes down one of the slaves will take over as the master. The new master is decided by an election among the slaves. There is two types of replication asynchronous and synchronous, asynchronously is when the master is updated and committed and then the slaves are updated each in turn, this could end up in data not being consistent over all servers. With synchronous data must be written and committed to each server and slave. The disadvantage with this is there is a read/write lock placed on the databases while this process is performed. Hope on Monday I will get access to some VMs so I will be able to replicate the replication process.

# Week 6 (27/Feb/15)

Blog<https://amulligant00122209.wordpress.com/>

I spent Monday devising unit tests to accompany, and ensure that the code I had written worked as expected. I tried a few different testing frameworks like FsCheck a property based testing framework. With FsCheck random data is generated and passed into your unit tests. These can be anything from an int to more complex types like generating random email addresses. There is quite a bit of work in setting up something like generating a random email address but once you have this done once it’s something you could call upon for ever more. It was because of the work involved I didn’t persist with FsCheck. I decided on using NUnit, this is a framework which I had used previously with C# and Java so this was pretty familiar to me. This framework uses the idea of asserting a result against an expected result. The result being what is returned from the function you’re testing and the expected result being what the function should return.

I had my project pretty much finished by this stage, so I decided to look at some extra functionality. As PostgreSQL has the facility to store Json I decided to store all the data in the invoice inside one field inside a table. These invoices could then be accessed by its unique invoice number. An excel sheet would then be populated and saved to a file for the invoice.

On Wednesday I looking into an F# concept call type providers. Type provider give the programmer method of accessing properties of data like (Json, xml csv or http) which have uniform structure. In the case of json for me I give the compiler an example of the json in the format I expect it in. I can then call on this example to access the properties of the json I actually want to use. Type provider do away with having to trawl through strings and find instances of words and so on. Type providers also give you a list of the properties in Intellisense for easy access.

On Thursday I spent the majority of the day preparing for my presentation on Friday. I had kept notes of all the technologies which I had used so it was just a case of re-familiarising myself with all of these technologies (F#, ZeroMq, PostgreSQL, Linq, Type Providers and NUnit).

Friday the day of the presentation which was supposed to start at 10:30 but there was problems with hooking the projector up to my workstation so it was re scheduled for 12:00 on one of the company laptops which the projector would work on. After pulling my project from git and downloading PostgreSQL to the laptop and inserting all the data needed, my programme ran once and then wouldn’t run any more. So after trying to remedy this problem without any conclusion. The presentation was again re scheduled for 3:30. I was pretty nervous and stuttered through the start a little. But I feel after that I managed to answer any questions they had to the best of my ability. After the presentation I gave a demo on my workstation, withal the software developers gathered round. I still feel my presentation skills need to be improved and it’s something I’ll have to work on immediately.

# Week 5 (20/Feb/15)

Blog<https://amulligant00122209.wordpress.com/>

On Monday I finished up writing little bits of code to tidy up my application.  I also started to refactor my code, changing my variable names or in F# value names to something more meaningful. I have a habit of calling values x y and z to check if my code is working and then change them to something more meaningful later. I think I should really spend another few minutes thinking of value names and not 10 minutes trying to figure out what they are and do at a later.

I continued with refactoring in Tuesday. What I didn't realise about refactoring was that it also includes refactoring the code. Ie (make conditional more efficient).

Wednesday was spent shadowing David O Shea (database administrator) David step me through the process of deploying software updates through different environments like UAT, production or qa. This looked like a very complicated process which David make look easy. He ensures me that's it just running the correct patches and scripts in the correct sequence. It is also his responsibility to fully understand all the changes which have been made even if it was James his pair programmer which implemented the changes. If any major changes had to be deployed a demo of the latest iteration would be required to bring the other software developers up to speed.

On Thursdays software developers are given a half of company time for dev training. The developers are expected to give a half hour of their time as well. dev training is taken in turns by each of the developers. The topics are usually something beneficial to the company. This week’s dev training was given on the topic of basic linux commands by Alan Murphy.

Friday morning is when I meet with my mentors David and Thomas to discuss what I have done over the week and hope to achieve next week. We discussed adding unit test to my project and saving my invoices as Json to Postgres. Also discussed was my project presentation next Friday. Where I will give a demo of my invoicing application and answer any question anyone there may have.

# Week 4 (13/Feb/15)

Blog <https://amulligant00122209.wordpress.com/>

Monday was spent looking at the looking at polymorphism in F#. I was researching this as I wanted to try send many different types of messages in my project. So for example sending a message of type Invoice and a message of type Order. I wanted to try put all these base types under the interface of message to be serialized and sent across ZeroMQ (more in the blog). After lunch I shadowed David and Marta who are now concluding a static website which the company use to demo their products to potential customers. They were preparing for a presentation which they were giving on Tuesday. I picked up a few tips regarding this process, one thing at a time don’t be jumping from topic to topic, preparation and admit you don’t know if you don’t.

Tuesday I began to look at Language-Integrated Query (LINQ) this is a set of features introduced in Visual Studio that extends powerful query capabilities to the language syntax of F#. LINQ uses standard, easily-learned patterns for querying and updating data, and the technology can be extended to support potentially any kind of data store. I hope to use this to query all the data I retrieve from the database

Wednesday was spent shadowing Continuum Commerce architects Billy Stack and Noel McGrath, I didn’t manage to get to see them writing any code but I did get to see them thinking about code. Billy and Noel where designing a document for product which they are working on. This is the same process which we used with Cathryn Woods in our software engineering project in second year. I still find it difficult to write these documents before coding but today the two lads really helped explain why this has to be done. Instead of me as in my software engineering project having to understand what’s going on it’s an entire team that need understand the process at Continuum Commerce.

The company uses a pair programming philosophy this is an agile software development technique in which two programmers work as a pair together on one workstation. One, the driver, writes code while the other, the observer, pointer or navigator, reviews each line of code as it is typed in.

I spent all day Thursday working on my project, I made good progress during the day. I set up a new GitHub repository so I would be able to work on the project from home as well as use it for version control. For my project I have a lot of the code written but it was written in section, I used these sections to test the components individually before integration these modules together today. GitHub link - <https://github.com/Alanmugz/InvoiceApp>

Friday

# Week 3 (6/Feb/15)

Blog <https://amulligant00122209.wordpress.com/>

Monday I spent the day with developer Gavin Pullen, Gavin usually performs tasks for the business side of Continuum like generation statistics. Gavin has access to the entire database system where he applies endless stored procedures to build up a portfolio for the business department. He performs certain procedures on a daily, weekly and monthly basis. From these reports Continuum are able to tell what products are working to their potential or if customer is not using the product correctly. Gavin showed me the full array of tables involved in all the product. Some schemas included as many as 50 table. Again this gave me a little more insight in the how Continuum works.

On Tuesday morning I had a meeting with David and Thomas on the progress of my project. We decided to change the database because a document store (MongoDB) would not work for the type of data they were giving me to use during my first project. So we decided to use a relational database by the name of PostgreSQL. We also finalised exactly what the project is going to do and how it will be done. I will add this to the WordPress site.

On Wednesday I worked mainly on the UI for my Invoicing Application. As F# dose not provided the same features as C# dose were you can drag and drop controls (buttons, labels, textbox’s) on to the form. All the code for each control has to be written individually. The code is not dissimilar to code which is automatically generated on dropping a control on to a form. This gave me a good reference when written the code for the UI. At Continuum UI are not really used so I didn’t spend much time making it look pretty. As soon as I was able to capture button click events and saved the data on the textboxes, datetimepickers and combo boxes I didn’t make any more changes to that code.

On Thursday I started off by taking a closer look at PostgreSQL, this is a relational database based on Oracle syntax. It incorporates many of the fundamentals which I have been thought in college. With the syntax being so parallel to Oracle it didn’t take long to get up and running. I was provided with actual records of transactions from the company’s data server. This data comprised of over 80,000 records. I will uses this data to perform calculations in my project. After lunch the entire company’s staff including myself attended a meeting about a new product. This meeting comprised of details of a high level overview of how and why the new product is to be created. There was also a lower level explanation of the technologies which were to be used and the reason for each of these technologies. This was very insightful for me again giving me more awareness of how the business works.

Friday was fairly uneventful, a presentation from HDip student Noreen Ward in the morning. She hopes to use Xamarin to develop an android application. This is something I would like to do in the future, Noreen thoughts on using this method of application development over the native means was very interesting. In the evening I managed to get a chance to talk to Billy Stack about the F# language. He is really interested in this way of programming and provided some valuable insight. And from my research I see more and more languages are becoming more and more functional orientated.

# Week 2 (30/Jan/15)

Blog <https://amulligant00122209.wordpress.com/>

Week two began with a meeting with David and Thomas about my project. I filled them in on what I had thought about during the weekend. And it was decided on the next steps I would take to progress. The next step was to continue with my research and form a code skeleton for phase one. This was the objective for the remainder of the week

The code for phase one is to be made up of two F# applications one a client and the other a server with a messaging broker in the middle (ZeroMQ). The client application will allow the user enter a decimal value and country currency which to convert to and the country which to convert from (amount they would like to convert, Country currency which to exchange to, Country currency which to convert from) this message will then be forwarded to the server where an exchange rate for that county the user has chosen will be applied. The exchange rate of the country and the total value of the exchange will be returned to the client application.

Day two started with a progress meeting with David and Thomas, the meeting was only a few minutes. I continued with my research into the technologies I am going to use. I also made a few amendments to the programs I wrote yesterday. I added the ability to be able to send objects via serialization between the two applications. I meet with David O’Shea CCS Head of Merchant Operations, he gave me an overview of all the CCS products like DCC (Dynamic Currency Conversion) and MCP (Multi Currency Pricing). I think this will really help me with the daily stand up meeting to understand all the acronyms the software developer’s use as well as giving me a basic understanding of how the CCS products work

Day three was relatively quiet, just some more research into ZeroMQ and F#. Didn’t really have any interaction with my mentors as everybody in the office was extremely busy today. A major company which CCS provide software are in town to conduct meetings on products which they are provided. The company is also expected to visit the office tomorrow.

Thursday I finished up on my initial ZeroMQ research. I compiled a document made up of the main points and features which I thought where relevant. This document will be used for myself as a refresher before my final presentation of my project after week 6. It will also be stored in the companies documentation archives if it was ever needed at a later date. I spent the second half of the day shadowing David and Marta on a project they had been working on. They had been working on a demo website which the company use to demonstrate their products. It was a static website with information hardcoded just for display purposes. The website was written in html, CSS and JavaScript these where some technologies which I feel I was on a level playing field with David and Marta. I was able to add my thoughts on the way thing where done. It also provide me with another insight into the way the company works.

Friday was spent researching F# and advancing the scripts I had written already by using more functional programming method and no the OO method I am so used of. This took up the best part of the day. It was tough to do thing functionally that I would find second nature in Java or C# but this will all come with practice.

# Week 1 (23/Jan/15)

Blog <https://amulligant00122209.wordpress.com/>

This week I began week one of work placement with Continuum Commerce Solution situated in the Kerry Technology Park as part of the second semester of my third year of my Software Development course with IT Tralee.

My first day began with an induction period where I was introduced to the members of staff and given a brief overview of the different sections of the company such as IT support, Development and management. The work placement plan was outlined which included an individual project for the first 6 weeks, a group project was also proposed but as I am the only work placement student I am yet to find out what the second six weeks will entail . Just before the tea break all members of the staff meet for a stand up meeting, here members of all department describe what they have achieved yesterday and hope to achieve today. Each member of staff only talks for a few seconds.

On the second day, developers David Murphy, Thomas O Connell and Marta Doberschuetz gave me a choice of new technologies to research and ultimately pick a project to develop from these or any others I might like to use. I spent all morning and afternoon researching F#, RabbitMQ and MongoDB among other technologies. After lunch I was assigned to spend the remainder of the day shadowing David O’Shea working on “Reporting”. He showed me the magnitude of the work involved on a large scale database. He was very insightful and gave me an excellent overview of how the system worked.

Day three started with liaising with David, Thomas and Marta about the technology I was going to choose and give a brief description of the application I was going to implement.

Day four started again with liaising with David, Thomas and Marta to establish if I had a better concept of what I was trying to achieve in terms of my individual project. I showed a better understanding of F# and outlined my desire to move away from OO concepts and try functional programming concepts. I also completed my first F# program to talk to a MongoDB database.

Day Five consisted of security training, this consisted of myself and new H Dip student Noreen being give a power point presentation and watching a video on security and social engineering. This consisted of strict rules we would have to adhere to.

At half 10 I had to give my presentation of the project I was going to implement, this took place in a meeting room. About 8 software developer attended the presentation. I described the concepts of my project. All seemed happy with my choices of technologies and ideas I purposed. But the senior developers want me to look into technologies which the company may benefit from. So I decide to substitute after a discussion to swap RabbitMQ for ZeroMQ and MongoDB for Redis. This didn’t change the overall architecture of my project. Just added a little more research. I will add a clearer overview of what the project will entail after final sign off of use of the two new technologies (ZeroMQ and Redis) on Monday.

I have noticed a strong emphasis on the staff improving their knowledge on existing and new technologies. Software developers take part in group learning activities like Book Club meeting where a chapter of a programming book is read and then discussed each week, dev training consists of each member of the software development department giving a presentation on a technology they are currently using outside of work. This is certainly a company moving in the right direction. I enjoyed my first week at CCS and look forward to the second week.