Self-Evaluation

To start my self-evaluation I worked through all the problems and successes and difficulties I went through with doing all the code and setup, in this document I will evaluate myself critically and without bias(to a possible extent).

First of all noted that my Coding for controls, what I mean by this is that when using drag and drop and using code for the controls I tend to write code here and there in bits and pieces when I get ideas. This is a problem because in the long term I tend to forget what I was working on and this results in some of my code is a copy of another control and extra debugging takes place. For example while writing the code on the controls in assignment 3 there were a few textboxes that worked the same roughly and a method could have been used, instead I went from event to event using the same code over and not realising I could have made a method because I was to focused on getting all the controls working at the same time.

Usability of the assignments are questionable, while because I designed it I know exactly how to use it all and know exactly what to do and what will break it outside users do not, but testing was done for the usability and mostly the usability was good enough. Most points were I am lacking in this is my exceptions. While I think the design and flow of the gui's are good users may find themselves making errors that cannot be fixed, so in other words my programs have low safety. But the further along I went the better I learnt to catch exceptions and by assignment 3 my code was better in terms of exceptions.

During assignment 1 I found myself extremely worried due to the parent child classes which I didn't understand fully. After a lot of messing around and research I finally figured out how the parent child relation is properly implemented and I got used to the code needed to do the classes. During this time I also discovered how useful breakpoints can be during debugging, e.g. while preparing my assignment I was filling out the user details into an array list but there were no values entering it, so I used a breakpoint and went through step by step to find that I had an error in my object creation, without breakpoints I may have never looked at that piece of code because I assumed it was correct and there were no syntax errors.

I struggled a lot with database linking. I was not used to the code but familiar with the wizard, at first my only problem was to figure out the code needed to do the database linking because I could not use the wizard due to the assignment restrictions. After a bit of reading I figured it out but I had no idea how to get the databases connection string, so I used the wizard and fount that I could locate it there and copy it over. After a while of coding the assignment the database opening reading and writing became easy and very familiar.

One of my biggest issues at the beginning of the year was time management, I didn't manage my time for my first assignment very well and ended up at a dead end a week before due date so the program suffered by me rushing through. By the time the second assignment came I was much better prepared in terms of my time management, and I did class diagrams and pseudo code for the events. I found that by having an outline of my program in front of me that had no actual code but had direction and order made coding the project much easier and less confusing now understand that the biggest part of any program is the design phase, changes can be made instantly and easily before implementing any code and all final decisions can be made which can save a lot of hassle.

During the year after the first semester I was getting discouraged that I didn't want to be a programmer, I thought maybe I lacked the interest or the capability of doing so. After a discussion with my lecturer it was put to my attention that it is not a lack of interest with programming but a lack of interest with the work that was given. After the second semester had started I had found all that he said to be true, the assignments given to us in semester 2 were amazing and fun to program, because all of them required the student (me in this case) to design a program that just had to have a specific function e.g. assignment 2 asked me to make a program that used databases and allowed me to add, remove, edit and show reports, the rest was up to me, I got to create a program that did what I wanted it to do in my case I designed a program that kept player data of a game that I play competitively (StarCraft 2) and coding and designing the interface was fun because it was what I wanted to do. More recently I've been working with an arduino and it is an indescribable feeling you get when you write code that you want to and you have the drive for and it works.

My skills with the interface design is getting better, first semester my designs weren't very appealing or user-friendly. The subject Human Computer Interaction (HCI) really helped me with conceptualising design by teaching me the different types of interactions in depth which also has broadened my ideas for after leaving college, by introducing me to different design concepts and techniques and different ways to test my projects I have learned how to better my programs interfaces and user friendliness, for example on my assignment 3 I used the natural setting evaluation technique by setting up a few questions and guidelines to follow while the participant interacted with the program, I noted some of her actions and faults in the program eg there was a reports dropdown that you would select and then choose a report but the buttons above didn't really let the user know that they needed to be filled in and the mistake was made that she didn't fill them in, so I rectified that and a few other design issues that made the program a lot easier to use.

One thing I want to note about software development that I've learnt, it completely changed the way I think about almost all the decisions I make even while not programming, its as if im coding my decisions I think about what the situation is and think of either result and when you think about it like that programming makes perfect sense

Earlier in the year I had a big problem with all my formatting I didn't have correct tabs and indenting, during assignment 1 it got to a really bad point and I had to start over, luckily it was in the early stages but after I restarted I focused a lot on my tabbing and spacing, it took a little bit longer just to get used to it but after I had finished it was much easier to proof read and spot errors. I realise how important it is to tab and space properly and it's become like second nature to tab correctly.