# Evaluation

## Evaluation of project plan

I felt as though the project plan was absolutely integral to the success of the project. The creation of table and form designs took a significant initial investment of my time however it also meant that when implementation began, I had a solid set of designs from which to work from. When developing in Visual Studio last year I often found myself lost because my designs were lacking somewhat, and the ideas I may have had for my project weren’t fully thought through. Learning from that experience I invested a lot more time into my planning stage this year, and once development began the rigorous preparation stage payed off almost immediately when I was faced with creating tables. Normally, I’d have had to guess at what fields and tables I would need and make up for mistakes later, but as I had tested and planned these tables and their functions throughout my planning stages I knew exactly what I would need in order for the system to work properly, which provided me with an extremely streamlined development process. My project plan had several stages including;

* An overview of the task at hand
* The development of user requirements with the client
* Extensive research into the available programming methodologies to use
* Creation of a plan through Microsoft project including a GANTT chart
* Creation of test data/rough ER diagram followed by a final ER diagram
* Creation of a Use-Case diagram
* Table designs including test data
* Form designs including property tables and some pseudo-code
* The testing of my designs against use requirements to ensure they fully satisfied all agreed requirements, and
* A test plan, which outlined all the functions I would have to test once the system was developed

This plan was certainly extensive, and took a very significant amount of my time, but each section of it proved integral to the success the production of the system; it both assured the quality of the final product by creating a blueprint created in close conjunction with the client, whilst also saving time and money by streamlining the development process in such a way that I was essentially following my own previously outlined instructions, with the only real issues I encountered being in the learning curve associated with linking SQL with C# and database I/O with the user (which will be elaborated upon in later sections of this evaluation)

Overall, each section of the plan can be directly linked with at least one instance of beneficial utility later in the development process:

*An overview of the task at hand:*

This one was an obvious necessity; I read the case study provided and highlighted they key weaknesses in the Mitchel Music administration system. I then carefully chose which section would be the most important to approach. I chose to develop the rental section of the system because the most money was involved in the school’s constant instrument rental and damage issues in single instalments. I then summarised this in this first section before continuing on to the next section.

*The development of user requirements with the client*

I’ll keep this section brief given that a much more specific insight into this section is provided in the next section of the evaluation. In terms of planning however, these requirements formed the crux of my system. They provided me with essentially a checklist of features to develop, which then automatically shaved the rest of the wealth of the features the system could have contained. Having said that however, my first attempt at requirements were created before I had a vast knowledge of SQL and database systems in general, so a few of my requirements were created on a false pretence of what and wasn’t possible, particularly in terms of scope with the time given, such as an original inclusion of development in the payment section of the system, where I would include cards and an ability to process cheques. It turned out that a professional instance of digital commerce is quite a complex system and a large industry on its own. In a slightly later redraft I chose to exclusively record the cost of payments associated by person, leaving the actual processing of currency to the people at Mitchel’s Music.

*Extensive research into the available programming methodologies*

Once again this is a topic which is addressed in a greater detail in a later section of this evaluation; namely “Evaluation of approach to solution”. Overall however the few days spent researching these approaches proved highly beneficial throughout the rest of the development process. Adhering to any methodology over a freelance approach ensures that all factors of development are considered and these guidelines relieved me of a potentially time consuming and dangerous task of figuring out the best approach myself.

*Creation of a plan through Microsoft project including a GANTT chart*

This section had an initial learning curve of learning to use a new software, in Microsoft Project. Having said this however, it also proved very useful to me. It gave me time periods in which to complete each section of the system and as long as I followed those guidelines I was guaranteed to finish the system in the agreed time. The GANTT chart seen in this document is my second attempt at a GANTT chart following a first attempt which was still relatively strong but featured less break down of tasks and gave me no window for overlap, which seemed like a good idea at the time but just resulted in ambiguity of task completion and strict time windows. The freedom within some sections (particularly in the implementation section) proved invaluable later, and this is reflect in the included iteration of my GANTT chart.

## Evaluation of user requirements

Testing of Designs against User Requirements

1. *The System should be able to keep information on each instrument available for rental*

A database file is used and maintained which then holds the information so long as the file isn’t manually deleted

1. *Any person wishing to rent an instrument has to be a registered as a student with the school*

On the rental form, a student ID is a required field meaning only registered students will be able to rent.

1. *Students may only rent instruments that they are taking lessons in*

The rental form instrument selector will automatically filter based on the instrument associated with the selected students instrument of choice.

1. *The System should be capable of containing details of all students.*

There is no maximum capacity for the database and it is dependent upon the storage space of the computer it is being used on.

1. *The System should keep details of each rental.*

A database file is used and maintained which then holds the information so long as the file isn’t manually deleted

1. *The System should be able to create a document that draws information from the student and instrument table in order to create a signable agreement for the rental of instruments.*

-Will return to this when I’ve completed my report

1. *The System should have a record of each payment made to the company from students pertaining to rental*

The payment table and the ‘view payments’ form will allow the user of the system to view all payments that are made to the company.

1. *The System should be able to identify who hasn’t paid their instrument rental fees, and contain a method of contact to request this payment*

The ‘fees paid’ field of the rental table will be automatically verified by cross referencing the payment tables records with the rental ID’s associated. For example, if the payment table lacks a record associated with the rental ID being verified, then its ‘fees paid’ would be set to ‘no’.

1. *The System should be able to record damages and repair costs.*

The repairs table accommodates this

1. *The System will have a method of storing information on the stockists that provide their rentable instruments*

A stockist will be associated with each instrument complete with contact information, seen in the designs.

1. *The system must give students an option to renew the rental of an instrument after one year.*

Once the date specified for return on a given rental record is reached, the user will be prompted to either renew this rental (Thereby extending the return date by a further year) or else marking the instrument as returned.

1. *The system will automatically request replacement instruments from stockists should an instrument be damaged beyond repair during rental period, or should the cost of repair exceed the price of a new instrument*

When damages are added associated with an instrument, all other cost of damages associated with these instruments will be added up and if it is greater than the cost of a replacement instrument then the system will automatically request a replacement instrument from the associated stockist.

## Non Functional

1. *Ensure a comprehensive set of designs are created in advance of development*
2. *Use an easily readable font*
3. *Ensure the system has minimal bugs and is appropriately tested*
4. *Use complimentary colours and colours that aren’t harsh to the eye*
5. *A log in system for security purposes*
6. *A user friendly, self-explanatory UI so that people who aren’t used to using a PC for administration won’t have a large learning curve*

## Evaluation of approach to the solution

The approach I took to this solution was the waterfall methodology, which in many ways worked effectively to my benefit, however fell short in some areas. I think the major positive with this approach was that as an inexperienced developed, waterfall provided me with a series of steps to complete in an order to ensure success. Other approaches such as extreme programming may have been more time effective, however the lack of structure may have left my work unprofessional. One advantage of iterative methodologies over waterfall would have been the use of timeboxing, which may have helped me with time management, however once again my inexperience leaves me glad that I chose waterfall because otherwise I’d be unable to take extra time overcoming difficulties in the systems production.

## Evaluation of solution

## Evaluation of own performance

When I began this project I had limited experience with C# and object oriented development, as well as no real experience in integrating it with other languages and no experience whatsoever in SQL. This project demanded the use of a user friendly object oriented front end system which consistently linked with an SQL database which was a huge step up for me but likewise one which is infinitely useful in the industry. It would be naïve at best however to say that it was only in the implementation of this system that I developed personally; the planning stage was excessive too, and I learned how to use the Microsoft office project software wherein I created the Gantt chart seen earlier in this document. Not only in software development but in wider life time management is absolutely paramount to success, and structuring an organised plan with time-based subsections is very good practice and something which both ensured my own success in this project but also improved by time keeping capabilities in my other subjects and extra-curricular activities. The planning stages also saw me utilising a Use-Case diagram for the first time and began my research into UML. I didn’t understand what normalisation was, however this task gave me an opportunity to learn the process and create a realistic normalised data set to the third normal form in a realistic situation. Taking this normalised data I was able to create an ER Diagram, something I was previously unfamiliar with but quickly learned about. Learning about the ‘crows feet’ system and how they can be used to indicate the relationships between tables was enriching and aided in my understanding and use of data throughout the rest of the project.