**Evaluation**

**The Programming Language**

To create the new system for Digital Funhouse, I have used Python along with SQL. Python is a fast and productive programming language because its syntax is easily understood which means more work can be done in a concise manner. Python is also the vehicle through which the SQL computer language was used to create and use databases. SQL allowed me to store various pieces of information easily and to retrieve and extract information to be used elsewhere.

The most useful feature in Python that helped me create the system was the debugging feature. The debugging feature in Python allowed me to execute my code one line at a time and it allowed me to see the values that any variables were holding so that I can see why the error was raised and make adjustments to fix the code.

Another useful feature of was the built-in modules provided by Python. I have used the ‘random’ and ‘datetime’ modules from Python so that I can easily generate random primary keys and to create dates to store in the databases. I have also used Python’s built-in SQL module in order to use and execute SQL within the new system.

When creating the system, I had to make use of validation techniques. With Python, I was able to implement validation through the use of error handling statements. With these, I was able to execute a certain piece of code whenever a user makes a mistake such as supplying data that does not match the specified parameters.

The ability to create functions within Python is a feature that my system is mostly based upon as I have used functions to store the essential code that allows the user to see the menus and to perform each task displayed. With these functions, the user can access whichever menu they want to access as many times as they want.

I have used other small features of Python such as loops and conditional statements. I have used loops to iterate through records in a database and to execute code again without exiting the system. I have used conditional statements throughout the system as a method of identifying which tasks the user wants to do.

**Comparison with existing systems**

I have looked at inventory management systems such as ‘Netstock’ and ‘Zoho Inventory’ to understand how other systems are programmed to manage inventory. Comparing the inventory management features of my system, they are lacklustre compared to the systems mentioned. My system stores information about the inventory in databases, makes deliveries for more items and it can create basic statistics and reports about how well an item is selling whilst the other systems can perform these functions more swiftly and they also offer other advanced features such as dashboards, forecast generation, automatic deliveries and barcode scanning. These features would appeal more to the user.

I have also looked at systems that deal with rental purchases such as ‘EZRentOut’ and ‘Rentware’. My system allows the user to manage customers who have made rental purchases, notify customers about their item and create invoices. The other systems offer these features alongside a surplus of other features such as rental price management and inventory management and their systems are more advanced and they allow the user to perform these tasks with ease.

Whilst looking at other systems, I came across finance management systems such as ‘Financial Force’. These systems offer innovative features such as creating financial reports, budgeting, forecasting and cash management. My system is limited when it comes to financial management as my system can only create reports and generate statistics regarding the expenditure of the business at a much smaller scale compared with this financial system.

**Improvements to the system**

If I were to be given more time to create the system, I would have considered implementing a graphical user interface as they are generally more user friendly compared to command line interfaces. Graphical user interfaces also allows the user to enter data into the given input boxes where text could be entered in any order rather than having to enter data systematically.

Given that I created a graphical user interface, I would have created dashboards that contained statistics about how much every item was selling and how much money they were making, highlighting the items that are popular. I would have also given the user more flexibility when managing customers who have made a rental purchase. Instead of having only automatic processes, I could give the users’ the ability to contact the customers through the system.

With a graphical user interface, I would have also programmed a way for the user to cancel a task at any time instead of making the user finish the task even if they accidently started it through the use of back or cancel buttons.

I could improve the system by adding more features to the system such as automatic deliveries, better cash management and forecasts. With automatic deliveries, the system will be able to make deliveries automatically when stock is low. I could implement better cash management by making the system able to access the financial information of the business so that it could identify the expenses of the business so that better financial decisions could be made. Forecasts could also be used by the system to predict the number of sales of each item by making use of other variables such as time, season and other trends. Forecasts could also be used to generate information about the likely financial situation.

**Self-Evaluation**

Throughout the making of the new system, I have been able to apply my programming skills in a real-life situation. To create the system, I had to plan thoroughly, thinking about what type of data I will be using and how the system will be programmed to fulfil the objectives. While the design document I created has been extremely useful, it did not contain everything that the system needed to do and there were also changes that had to be made in order to make sure that the system functions correctly.

When creating the design for the system, I was able to successfully think about the code logically and think about what the code needs to do and how it is supposed to do it. My ability to program helped me in making the system efficient as I was able program the system to perform certain tasks whilst trying to limit the number of lines of code where possible. My knowledge in programming came into use when testing my code for errors as I knew about the common types of errors that would be raised and what would cause them.

Before I started to create the system, I already had the skill to program in Python using the fundamentals of the programming language. However, I had to learn how to program using SQL as SQL is an essential language regarding databases and the language was necessary for me to create the system. SQL was easy to learn as its syntax is very close to the English language making it easy to remember.

When making the system, I had to manage my time so that I could finish creating the system within the given deadlines. I had to work even more because once I started to program the system, I realised that it was going to take much longer to create it due to how a simple task could require a lot of code to perform and due to not having a plan as effective as it could have been.

I had to think about how the user would be using the system during every stage of creating the system. I needed to use a lot of validation to reduce the chances of an error occurring. I also had to be precise when writing instructions for the users such as specifying the format of a date or a field name.

During the creation of the system, I first created an initial untested draft so that the program could have a basis which would make it easier to make sure that the system achieves all of the objectives. The initial untested draft inevitably contained errors but a programmer looking at the system would understand what the system was trying to achieve as the programming logic was present.

There was a lot of errors in the code and some of those errors took quite a while to solve but I essentially had to use the debugging tool and test the entire system line by line to make sure that it was executing perfectly. Most of these errors occurred either due to being uneducated about SQL or it was a complicated error that could only be identified by careful investigation of the code. I also had to ask for feedback from others so that I could identify how the programming could be improved and how certain errors could be avoided.

**Future Improvements**

If I were to make another system, I would make a better plan for creating the system. I would do this by describing the tasks the system needs to do in a way that allows me to understand what my system would be doing and how it could look like.

I would also learn more about how to solve more types of problems in Python as there were issues when creating a task for the system because I did not know how to program a way to perform the specific task which ended up taking more time for me to create the system as I had to teach myself about how to solve those types of problems.

Another improvement that I would make would be to test my program as I am creating it so that any error would be caught and dealt with straight away and it would cause less trouble when testing the system. I would not use the method of creating an entire untested draft again as I would create a plan that contains the basis of the program so that I could have an idea of what the system would look like.

I would also try to do a better job at managing my time as I now know that programming a system is not something that can be done quickly and it is going to take quite some time to program a task and to debug that task if any errors arise.