Evaluation

**Introduction:** In this part of the project, I will identify good features and shortcomings of my solution and I will also describe potential improvements that could be made to improve its effectiveness as well as specific changes of approach that I would adopt in the future to avoid the problems experienced during development. I will also be comparing the system that I have made with a commercially available system. Moreover, I will also describe how I found using visual basic as my programming language and the various different tools and techniques used whilst creating the program, provided by the Visual Studio IDE. Finally, I will evaluate my own strengths and weaknesses that have became prominent during the design and development of the system.

**Evaluation of Visual Basic as a programming language:**

Overall, whilst developing the program in Visual Basic, the intuitivity and simplicity of the language aided me significantly during development. Moreover, the IDE used (Visual Studio 2019) had a wide range of tools available that facilitated the creation of user interfaces and the implementation of event driven programming. For example, the use of forms, buttons, text boxes, grid views, labels, combo boxes and list boxes allowed detailed and user-friendly interfaces to be created.

The solution explorer allowed me to maintain all of the forms and to be able to switch between them with ease, as well as enabling the debugging of the forms individually instead of the program as a whole, to save time and ensure the intended functionality of each form.

The properties tab allowed me to assign meaningful names to the text boxes, buttons and other tools used in the form. This made developing each form easier as it was easier to keep track which textbox held which value e.g. a textbox called txtFirstName. This made the maintenance of the code a lot easier. Moreover, using the properties tab I was able to add ToolTips to certain text boxes and buttons, which would show when hovered over by a mouse. These tool tips would inform the user of what the current tool stores/does to enable a better understanding of the code.

The code for the manipulation of tools was very understandable, and consisted mainly of clicking on a button, and Visual Studio 2019 would instantly create a private sub-program which I would then enter the code for what I would like to happen when said button is pressed. Generally, the communication of the database with the program was simple and the tool DataGridView enabled tables from the database to be read with ease by customers and staff members alike.

Error messages produced by program where highlighted in detail by the IDE, usually giving a clear explanation to what was wrong with the code. This sped up the process of development and quality of life improvements. Due to the Visual Studio IDE series being popular, when an error was not too clear in relation to the course of action needed to fix it, a simple google search of the error usually yielded a plethora of results with subsequent fixes. In addition, in the case of the program having multiple errors, they were all displayed in the Error List and clearly specified which line of code of each form they were in, and if clicked, it would take me to that line. This also enabled an overall faster process of error correction.

Furthermore, I utilised the ability to write comments in Visual Studio 2019 to accompany code which made it more clear and understandable as it defined what each line of code was achieving. These comments have been added throughout the solution, one line before the line of code that is being described and are highlighted in green to promote readability. This was possible by utilising the syntax of the Visual Basic language which involves writing a comment by starting the line with an apostrophe e.g. (‘This code does this).

Debugging tools provided by the Visual Studio 2019 IDE were both useful and in abundance. For example, the use of breakpoints was available which allowed me to pause the code at certain points and examine the state of each variable and to then compare this with the intended value. This assisted me in being able to verify the correct functionality and execution of the program.

In addition, the component of the IDE IntelliSense supported the development of the program greatly. This code-completion tool delivers a great deal of functions such as list members, parameter info, quick info and complete word. These features helped me to further understand the code I utilised, keep track of the parameters used and to add calls to properties and methods with ease.

**Comparison of the system with commercially available systems: (QEX)**

I have visited and spoken to my local game store (refer to Investigation - existing solution to similar problem 2) to ask them about how their commercially available system (QEX) operates and how it compares to the system I have elaborated. I asked them about the main features of their system such as the ability to add/remove games to/from the system in reaction to customer purchases and games bought from customers. Moreover, the ability for staff members and customers to view all of the games in stock. Their system was similar to mine in the way that it stores all of the games they have in store in a database as well as the integral part which allows the store to buy and sell games. However, this game store does make use of hand-held barcode readers which works with QEX to be able to facilitate the process of data input.

Moreover, QEX has the ability to give store credit to customers for games they sell to the store. This amount is calculated from how much they would be given in cash, however is incremented to incentivise people to reinvest their money back into the store. This is a feature that would definitely increase the profits that Mongoose Games would receive if this was added to my system as re-investment in the store would be encouraged which would reduce the loss of the business from the reduction of pay-outs to customers.

In addition, QEX provides customers with the ability to leave reviews on games once they buy it, for future customers viewing that game later on to see. This allows customers to view past experiences to either motivate or discourage a purchase. This would benefit my system, as although it could be negative, it would overall create an idea of transparency for Mongoose Games and would make customers feel that they are more trustworthy and reliable than other stores, which provides them with a comparative advantage, which would further increases profits. The reason why my system cannot do this is that once a game is sold from the system, it is deleted, for this to work I would have to use a relational database with two interconnected databases to enable this form of feedback.

Furthermore, QEX also allows customers to access the system from home via an application they can download with ease. From this, they can view and search through all available games and subsequently set up sales for their games to the store. Moreover, customers are also able to scan the barcodes of games that they have using their phone camera, to find out how much they would get for their game without needing to input any data or make a query with a staff member. This would be a considerable improvement for the system I have created for Mongoose Games as it would facilitate the accessibility of the system which would provide convenience for customers, leading to higher satisfaction and participation. In addition, the fact that staff members wouldn’t have to be queried by customers constantly asking for quotes on their games, would promote productivity in the workplace and would give time for other actions, overall increasing efficiency of the business.

Moreover, QEX provides a feature which automatically sends replies to customers request to the customers email as well as their account on the system. This ensures that the customer receives the message in the case that they don’t use the app much or have forgotten about their query. This would be useful to have implemented on my system, although email addresses of customers are not stored, as it would remind customers to continue their transactions with us and would overall promote customer activity which is vital component of a thriving business and although Mongoose Games has 500 current active customers, this customer base would increase significantly with this feature.

However, the system I have developed for Mongoose Games includes the feature for customers to be able to view all of the information that it stored about them for example their address, phone number or username/password. Currently, this is not a feature included in QEX and I believe that having this feature in my system is essential as it recognizes the customer’s right to view what Mongoose Games as a company stores about them and adheres to this. It also enables customers to verify that all of their personal details are correct which promotes the assurance of data integrity of the database.

I have also spoken to my local newsagents (refer to Investigation - existing solution to similar problem 1) and the use of their new system (FRESH4US) and their ability for new editions of newspapers being automatically ordered is something that could be implemented into my system, in a way that when a the next edition of a game series (of a game that is currently in stock) is released, it automatically gets ordered from the warehouse. This would enable Mongoose Games to ensure that they gave the latest games in stock, which could provide a comparative advantage over competing game stores that would take longer in ordering new games.

Moreover, when food products reach their sell by date they are thrown out and subsequently deleted from the database, this could be done in my project, for when games are older than 10 years they should be deleted from the database as there is minimal chance they will be bought by customers. Additionally, it prevents the stock of Mongoose Games being dominated by older games which could, at first glance by a customer, look like that they stock old and outdated games, repelling customers from the store.

On the other hand, my local newsagents doesn’t provide any features for customers to add magazines / food items to their wishlist as my system provides (with games). This could put the FRESH4US system below the one I have elaborated, as it’s essential that customers are given the ability to inform the store of new items they wish to have, as it enables the store to cater their stock in order to adhere to the needs of their customer-base which is fundamental in a thriving business.

**Successful features and suggested improvements:**

Successful features of the program include the ability of Mongoose Games customers to set up purchases and sales of games as well as staff members to view these and subsequently attend to their wishes in the store. This will ensure that the interaction between staff members and customers is minimal unless an arrangement has been made which reduces clutter in the store as well as wait times and allows customers to carry out tasks more independently and efficiently.

Another successful feature of the system is the overall improved productivity of the business, highlighted from the change from a paper based system to an electronic one. This is a significant improvement as tasks don’t have to be carried on paper, which negatively affects the structure and professionalism of the business. Moreover, the everyday tasks are completed expeditiously and makes the processes more secure and coherent.

However, in order to make the program more efficient, adding features that are compatible with hardware such as bar-code scanners would diminish the need for inputs of a game’s details that would be identified instantly through the scanning of a bar code e.g. Price. This would in turn enable the program to operate more productively as more customers would be attended to per day regarding buy/sell requests. This would also increase the profitability of the business.

In my test plan I identified that the format of the data should be more checked when a new user is creating an account. However, more thorough measures should have been used to ensure complete validation. For example, it would ensure that their postcode entered is in the format LLN NLL or LLNN NLL as per UK format. This would guarantee data integrity of the database as stricter rules would be placed on data input as well as it would help guide the user to the format required which would also speed up the process for them.

Another improvement to the system that could be made would be to add text-to-speech capabilities. This would enable Mongoose Games customers with visual impairments with the ability to have text narrated to them e.g. to listen to what a staff member has said in their reply to their request or to listen to their original request to ensure it’s comprehensible. Furthermore, a speech recognition feature could be added which would allow disabled users to navigate the system and dictate their inputs. Overall, these features would make the system more inclusive and engaging for a wider range of users.

Moreover, notifications could be added to the system both for customers and staff members which would benefit both parties tremendously. For example, if customers would receive notifications after a game in their wish-list would come in stock, it would be beneficial to both them and the business, as the customers would be more likely to be aware of the game availability meaning it’d be more likely they would buy the game. An option to then customise notifications for certain things would also allow customers to prioritise their interests, if they would be more interested in knowing when their wishlisted game would come in stock than if a staff member would’ve replied to them saying they have updated their details. With this use of notifications, it would save time constantly checking their messages which would save time, improving customer satisfaction. Staff members could also benefit from this if they were to receive a notification once a customer sends a request in, to be able to attend to their request as soon as possible which would allow them to get more work done and improve the overall efficiency of the program.

In addition, a feature for staff members to specifically add new games to the system that would’ve been bought directly from a warehouse. However, although unprofessional, this can be done by using the add/remove game from system form, and choosing to add a game as being bought from a customer, entering a random customerID and then choosing not to record the transaction, likewise this can be done when removing games from the system that have been lost/stolen. Nevertheless, these incidents were not given as much importance as they do not happen on a regular basis and due to time constraints, the general process was prioritised.

Another feature that wasn’t included due to time constraints was the original user requirement of allowing the system to be accessed via an application so they could make use of the program on the move or at home. This wasn’t capable of being implemented into the system as it’s not internet enabled and the data the system works with can only be manipulated on one computer at a time. To enable this possibility in the future there would have to be an online database that would be connected to, and a subsequent compatibility of the program on different devices e.g. mobile, tablet for users to use on the move or if they don’t have a computer at home. However, with this, extra security features would then need to be implemented to ensure the protection of customer data from unauthorised personnel. This is because with the system being internet enabled, it would be open to hackers that could carry out malicious activities e.g. SQL injection.

Furthermore, with regards to general improvements that could be made throughout the forms in the current user side of the program, when a user is changing their credentials, they should be asked to type in their current username/password (depending on which they want to change) as well being obligated to enter the new value for the chosen field twice. This is a form of verification (double entry check) which would promote data integrity and reduce the chance of a customer changing their details to a mistyped value that they will then have to get changed externally. When customers decide to view their details, the fields displaying their information should be hidden and the value they contain should only be shown when the customer hovers over them. This would be a good feature as it would ensure that during the event of someone shoulder surfing a customer, attempting to steal sensitive data about the client, the customer is given time to check their surroundings before viewing information, or even if they do not detect a shoulder surfer, only one piece of information will be compromised at a time. Overall, this would improve the security of customer’s data to external factors. Also, when a customer sends a request to the staff for any reason, they should receive an automated message explaining that their request will be attended to within the next 24 hours. This would enable customer service to be given importance as well as giving the customer recognition and a feeling of importance as an essential stakeholder for Mongoose Games which would subsequently improve customer satisfaction and communication. In addition, when customers are viewing the replies that they have received regarding their requests, what would’ve been a good feature would be to allow the customer to see the date/time that the request was sent to them by a staff member instead of ordering the messages by “RequestID” which firstly, they might not know is chronological and secondly they might not know how to order the field. This is important as it will allow customers to understand when their request was attended to in order to have a record of the response, as well as to also give feedback on response times. Finally, when a customer searches for a game they currently have to know the title of their desired game 100% accurately. It would be more suitable if that they were to receive suggestions of games existing on the database depending on the current input - making use of real time updating. For example, if they type in “a”, they would instantly receive suggestions on games that they could be referring too such as “Astro” or “Auxilary Mountain” etc.

Additionally, with regards to general improvements that could be made throughout the forms in the current staff side of the program, for viewing customer requests, where if a staff member has just replied to a request made from a customer - they will be asked if they wish to delete the original request that was sent to the staff members, if yes is pressed then instead of the request being deleted completely from both the DataGridView and the database (as is in current system) it would be better if the request was simply hidden from the staff members’ view. This would achieve the reduction of clutter that deleting the request does, but also would enable the record of customer requests to be kept in the database. This is important as in the event of a customer making a request, receiving a reply and then not being satisfied with their response and so making a complaint - there would be no record of them making a request in the first place and so staff members could be oblivious to the situation and would have no way of reminding themselves of the circumstances. In this case, it would provide very poor customer service and satisfaction for the customer which could possibly have a negative impact on the reputation of Mongoose Games which would prove detrimental to the business. Moreover, when a staff member is updating customer details (or a customer is creating an account) and an address is inputted, currently on the system the only validation for this field is checking that it doesn’t contain any numbers. However, what it should validate is the fact that the entered address is an actual geographical location. If this were to be added, it would ensure that all customers addresses on the system would be actual locations which would subsequently guarantee data integrity. Moreover, it could aid the customer/staff member through auto-completion e.g. if the beginning of an address is entered, geographical locations would be suggested or, when a user is creating an account, if they enter an address - the postcode, county and town fields can be automatically determined by the system and use this information to populate the corresponding text boxes, decreasing the time taken for data input. Finally, when a staff member is viewing all of the games that have recently been sold/bought to/from customers, they should be able to see the date and time of when the transaction took place which would enable Mongoose Games to accurately pinpoint the time of a transaction for any reason e.g. proof of purchase/payment or proof of selling. Moreover, this would also enable reports to be produced e.g. all of the games bought/sold in the past week, enabling the ability to track the progress of the business and to evaluate profits. Furthermore, the staffID of the staff member who sold a game to a customer should also be stored in the database which would provide further value to these reports and would even be utilised in order to implement commission for staff members which would subsequently encourage them to sell more games which would increase their interest in the success of the business - which is what a healthy workforce should have. It would also enable the boss (my client Mongoose Hanks) to be able to give people salary raises depending on how many games they have sold to customers, further promoting rewards for contributing significantly to the success of the business.

**Strengths and weaknesses of my personal performance:**

At the start of the project, I was given a document that explained how the project would be structured and the different sections that would have to be completed in the given time frame. Whilst working on my project, as well as the code for the system, I used version control and backups to minimise the possibility of a loss of work which would’ve had the possibility of setting my progress back substantially which, when undertaking a large time-limited task, is very important to avoid. As a result of this, I didn’t lose any work relating to my project which was advantageous as it allowed me to utilise all of the time I was given efficiently.

Moreover, I believe I worked through the project with a successful use of time management, ensuring that deadlines were adhered to and that time was spent efficiently. An example of this was when I started to ensure that I was able to connect my database to the program 4 months before the software development was due. This was done to ensure that once I reached the software development stage, time would not be wasted on this area. This worked well and in retrospect, I’m content with the way I approached this project and in the time that I managed to complete it in.

On the other hand, I had limited knowledge of the programming language (Visual Basic) as well as of the skills related to connecting databases with programs. This is something that took a lot of research and troubleshooting to be able to overcome which in turn was very time consuming and was generally hindering to the productivity of the project.

In addition, the fact that I had to work on the project from school and home, it was troublesome having to constantly send over the files for the solution to be able to develop and run the program at varying locations. At times, this lead to compatibility and file transfer errors due to having a Windows 10 computer at home and a Windows 7 computer at school. I could’ve asked the school to allow me to access my student files at home to enable ease of data transfer, but since I didn’t do this, the way I instead did it inhibited progress at certain times and consumed a lot of time.

**Changes of approach that would be adopted in future to avoid problems experienced during the project:**

In the future, I would consider using a programming language such as Java that enables compatibility for a wider range of devices in order to avoid the problem I had trying to make my program work with mobile applications as well as on desktop computers which is extensively hard using visual basic.

Moreover, I would plan each stage of the project, including what I would be talking about and the necessary sub-headings, before plunging directly into the issue which is what I did, which led to a lack of structure and organisation which without doubt affected the time it took to complete the project as well as the content’s relevance to the project.

Furthermore, I would also change the order in which I would carry out the project. For example, I would conduct the Prototype of the program before the Design. This would enable me to get a more detailed insight to the functionality, possibilities, structure and uses of a select few user interfaces before designing the bulk of the program. This would allow the design stage to be completed under more informed circumstances of the projected solution, which would enable me to create effective and realistic user interfaces. This would in turn avoid the issue I had which lead to my design being poorly structured and imagined, providing less guidance for when actually creating the solution.