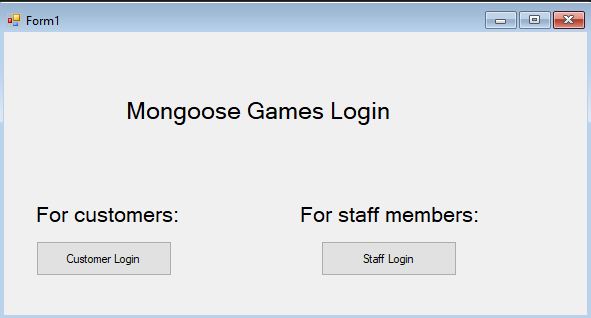
Prototype

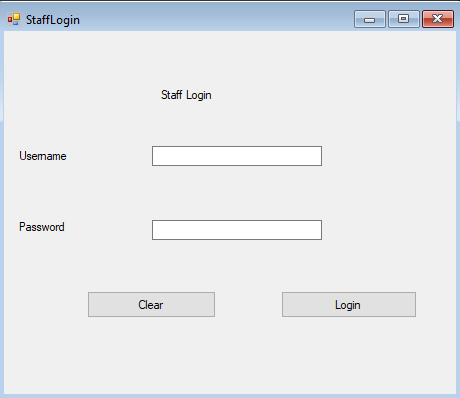
**Introduction:**

In this section of the project I will be explaining the parts of the proposed project that I will choose to prototype and the reasons for doing so. Furthermore, I will explain why I have chosen not to prototype certain areas of the project. This will then be followed by an evaluation and shortcomings and improvement section.

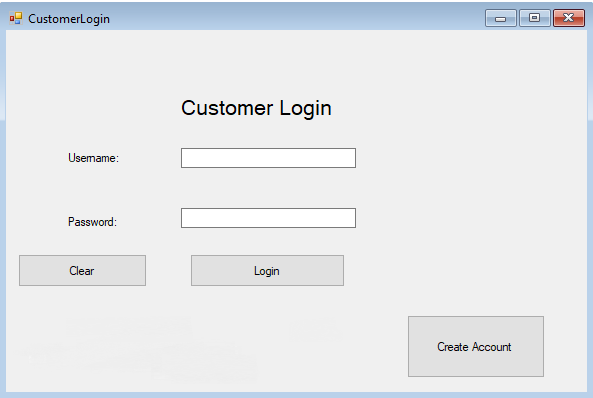


I have chosen to prototype the main login section which is the first thing any user will see when they log into the system. The login system that I have created works with a Microsoft access file where all of the user’s details are stored in. This ensures that only the people who have a prior created account stored in the database can be granted access to the system. During the creation of this prototype I was faced with various errors and issues as I had no experience with linking MS Access databases with VBA and so I had to engulf in fairly deep research in order to achieve this, but eventually I was able to create the connection resulting in a fully functional login system for both customers and staff members.

Seen below is the login that staff members will use to login to the system and access all of the functionality at their disposal

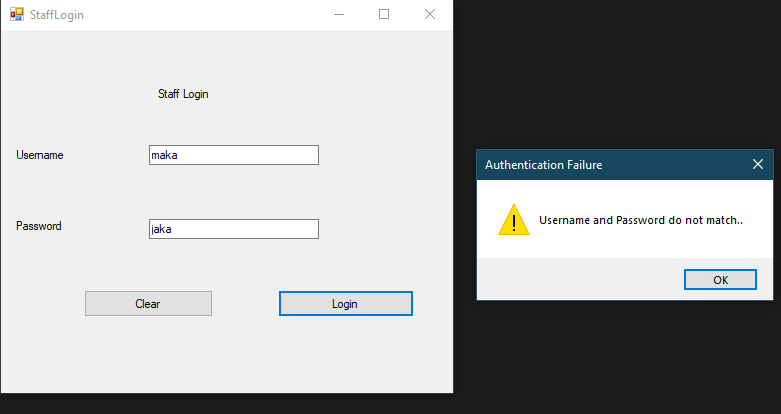


Seen below is the login that customers will use to login into the system and access all of the functionality they have at their disposal

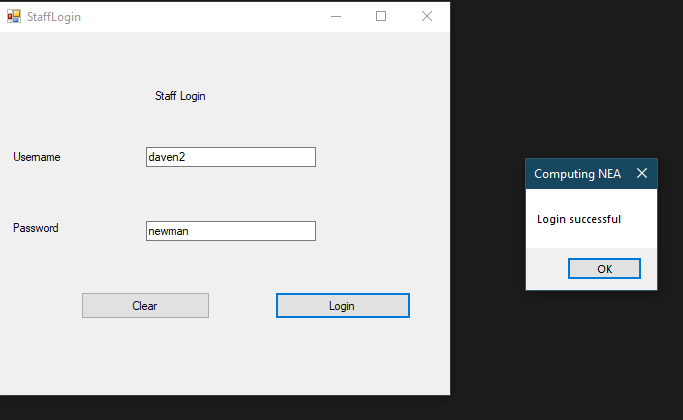


Output when invalid credentials are presented

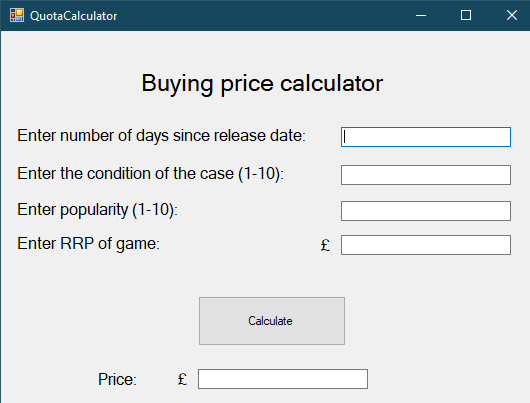
This image below demonstrates the functional login screen that I have prototyped for the user requirement specification. The prototype shows an error message dialogue box to confirm that the prototype successfully validated the user login.



Output when valid login is presented

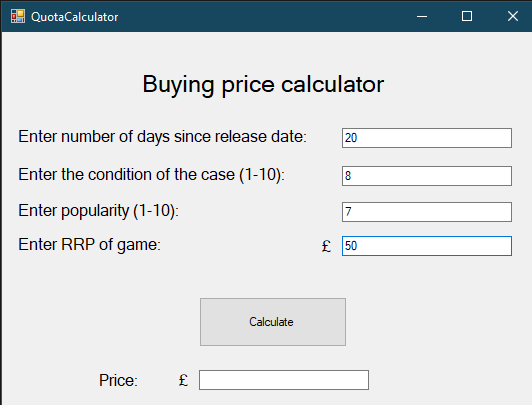


The next part of the system that I prototyped was the quote calculator. This is a facility that will be used regularly by the staff members in order to give them an indication for how much they should buy a game off a customer. This calculator makes use of a complex algorithm which accurately determines a price based on factors. The difficulties I faced while programming this part was that I found it hard to come up with an efficient and reliable algorithm that takes into account all of the factors of the game and casing that would ensure profitability and maintain the integrity of the business.



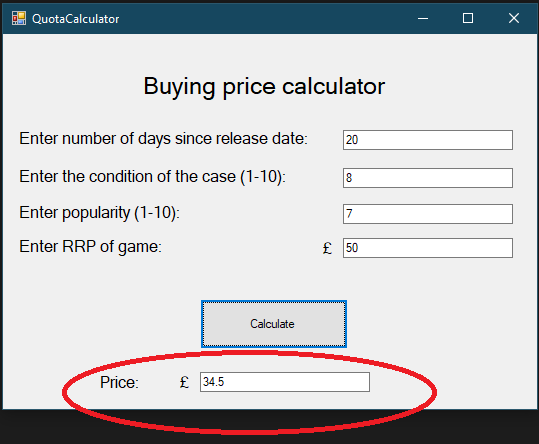
Test data entered into the quote calculator form

This shows normal, expected data that would be entered into the system about a game given in by a customer.

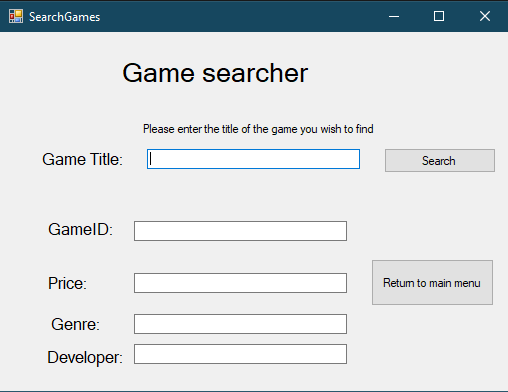


Output

This shows the result after all factors being used in the algorithm, to give the quote for the game.

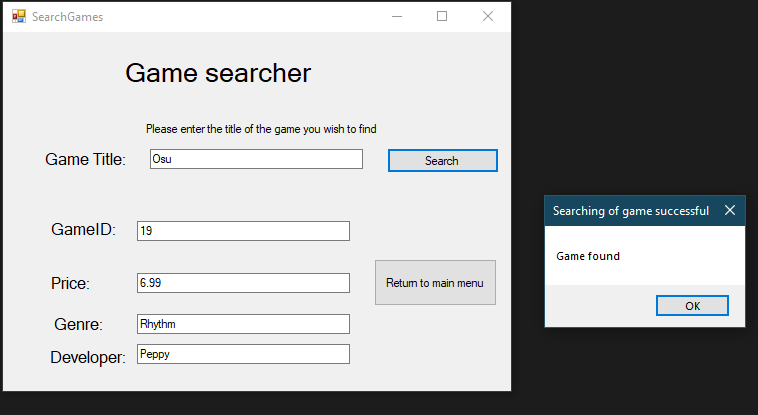


Furthermore, I have also prototyped the part of the system which allows customers to search for games by their title to see if their desired game is in stock. I was able to carry out this by again, utilising the pre-existing connection of my database with my program, however, the Structured Query Language (SQL) statement works with the Table “Games” and not “Staff” or “Customers” which I used for the login section.

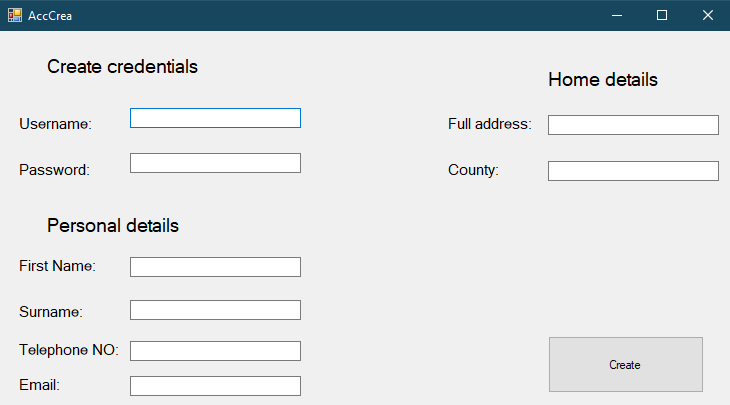


Test data entered into the form and output of the game’s relative details displayed via text boxes.

This shows test data being entered into the search bar for a game that exists in the database, upon clicking the search button, the database is searched and the data is instantly retrieved and populated into their corresponding text boxes.

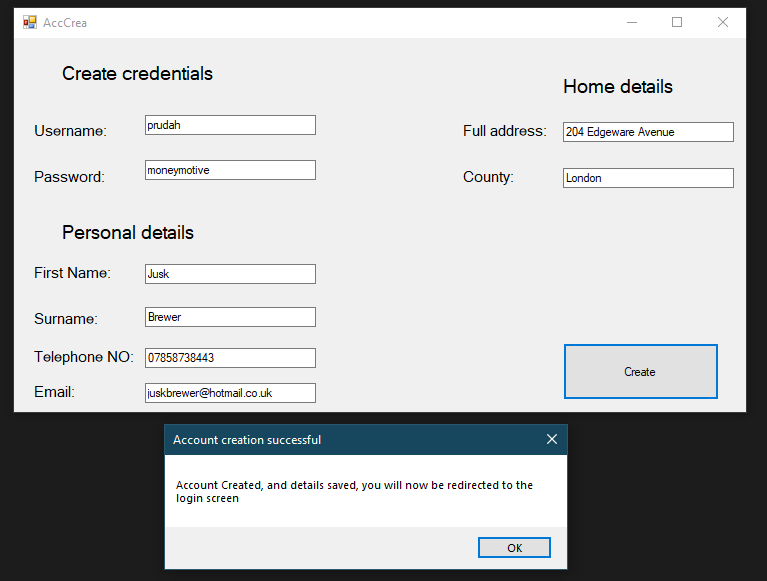


Moreover, I have gone on to prototype the “Create Account” part of the system which is used by customers who currently do not have an existing account on the system. This will be the circumstances for every single customer at one stage. This section consists of getting a full fact file of the customer including their address, personal details as well as their desired credentials that they will then use henceforth when logging in to the system.

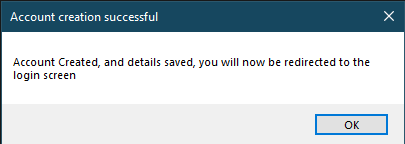


Test data entered into the form

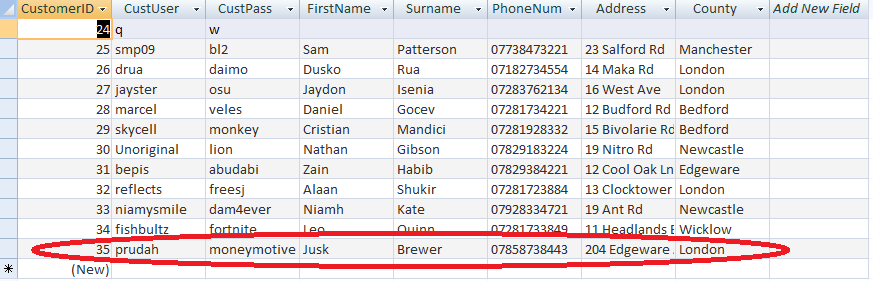
This shows a selection of data that has been entered into the system which would be done by a new customer who wishes to create an account in order to have access to the functionalities of the program. All of their inputted data is then sent to the access document in the corresponding tables, ready for future reference e.g. login



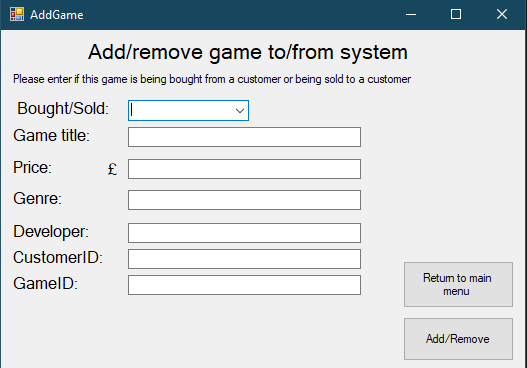
Output



Data saved in the MS Access Database

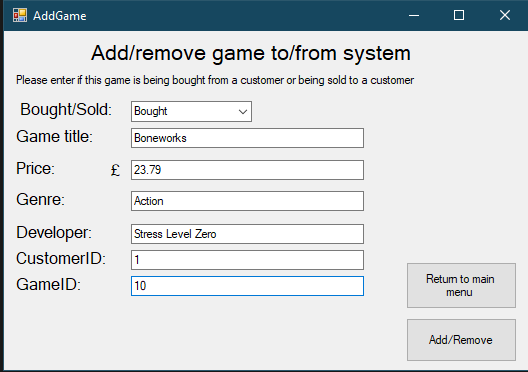


Finally, I have decided to prototype another integral part of the system which allows staff members to delete/add games from/to the database (see form below) depending on whether they have been bought from a customer (add) or sold to one (delete). The details of the game will then be subsequently added or removed from the Games table in the database and the transaction will be logged in the RecentGames table for when the staff member desires to see a report of the games that have been sold/bought recently.



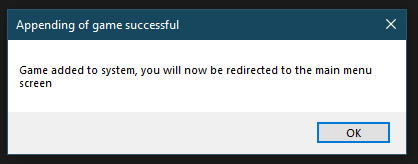
This form above will allow for a staff member to enter related data of the game, specify if the game is being sold or bought, and then which customer is involved in the transaction. These details will then be added to or removed from the database(see form below) and then the staff member gets a confirmation if they want to log the transaction into the system. It is possible that they won’t want to if they had to delete a game from a database from another reason than that of it being bought such as if it was stolen or lost. Moreover, if they want to add a game to the database that hasn’t been bought from a customer however has been brought brand new from a warehouse to then be sold in the store.

Test data entered into the form



Output

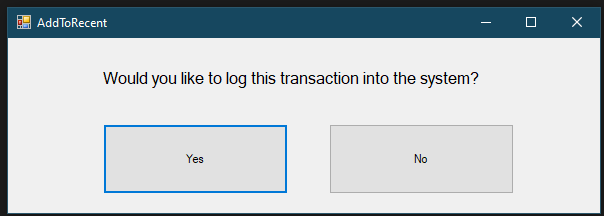
This informs the staff member in their success in adding a game to the system. They are subsequently redirected to the main menu.



Output (confirmation)

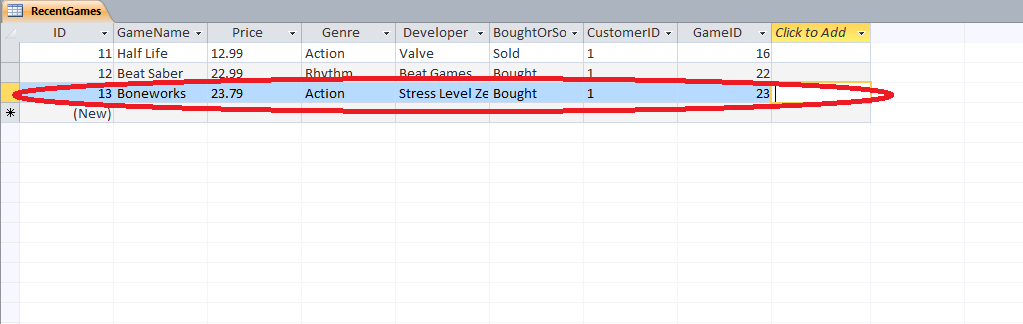
I added this confirmation for if the staff member wants to add a game to the system that has been bought brand new from a warehouse or remove a game that has been lost/stolen from the store without having to log the transaction to the “RecentGames” table which stores all of the games that have been bought/sold from/to a customer.

If the user selects the “Yes” button, then the transaction will be recorded in the “RecentGames” table (which records all of the games that have been appended/deleted from the system) as well as being added/removed from the “Games” table and if no is pressed the transaction won’t be recorded in the “RecentGames” table but it will still be added/removed from the “Games” table.



Transaction being recorded in the system

Below shows the “RecentGames” table in the database, confirming that the game has been successfully logged in the database as by the staff member’s wish, as a recently bought game.



Sections of the proposed solution I haven’t included

I haven’t developed the feature that allows staff members to view the recently bought and sold games from the store as it’s similar to the game search prototype as it reads from the database and produces a result, however in a listbox, which I am yet to find out how to do. Nevertheless, this will still be included in the development stage.

Moreover, I haven’t included the function that allows staff to update customer details in the program as that isn’t on high priority as at the moment only test customer data is in the database which won’t change. Yet, of course this will be implemented into the development stage for when this system goes live and real customers will be in the database which can have constantly changing information such as address, phone numbers and emails.

In addition, I haven’t coded the functionality which allows the customer to view all games that are currently available to be bought and also haven’t made it possible for customers to send buy request or sell requests of games, this is also due to how the system at the moment has no real customers but this will too be added in the development stage.

Evaluation:

Overall, the current prototype that I have established functions well and effectively carries out the tasks it was set out to do. It enables both customers and staff members to log into the system and carry out some of their respective tasks. For example, staff members are able to calculate the price of a game that a customer wants to sell which works hand in hand with the function that allows customers to request a quote for their game. Moreover, the Game Search feature of the project to be used by customers to allow them to search the store digitally for any desired games. This can also be carried out by staff members if they want to ensure games have been successfully added to the database or just for maintenance purposes. The account creation for customers has also been established which allows new customers to create accounts for future login and use of the system.

The program debugs with ease and works as expected. The current prototyped system has already solved some underlying issues that were experienced with the anterior paper-based system such as different handwriting of staff members reducing the accuracy of the data stored. However, there is still a long way to go before this application can be implemented on the newly installed computers at the store.

Shortcomings and Improvements:

I definitely need to add more validation to ensure that the data a user inputs meets certain requirements e.g. range check, length check, format check and presence check which will subsequently prevent erroneous data from entering the system . I should also consider adding verification to my system to check that data has been inputted correctly, such as using techniques such as double entry. These processes will in turn lessen the amount of issues caused when customers or staff desire to utilise the information in the database to either generate outputs or input into them.