2018

Evaluation

JACK BROWNE - 555784

An evaluation of the programming language used to create the solution.

To come up with a solution for this problem, I have used Visual Basic for Access (VBA). VBA is an event-driven programming language which is used mainly for Microsoft applications such as MS-Excel and MS-Access. VBA is very useful for quick implementation of a solution such as the one I have developed due to the huge number of in-built functions. These in-built functions allow for complex calculations and functions to be completed quickly and easily.

The effectiveness of the programming language for my system has been somewhat volatile. For some functions, such as lookups, the creation of a user log and table manipulation, the program has been easy to use. This is due to the in-built functions being quick to call and manipulate so that they can be used in multiple ways. On the other hand, there have been some situations in which I have found the program difficult to use and limiting. For example, I had to experiment with an extended workaround for finding an artist's act name based on their username in a SQL query. I had to create a function which would call upon a sub in which the SQL / VBA compatible variable was declared. I was unable to simply insert the variable name into the control source.

Access also has extremely effective user interface. All of the tables, forms, reports and modules are shown on the left side of the window, the current form is shown in the centre and any object properties are shown on the left. The system can intuitively switch between design view, form view, table view, report view etc – each with different visualisations and functions. Access can save all of the table manipulations. Some security issues are apparent, yet there are some ways that I have managed to secure the system. The modular nature and easy-to-view hierarchy of classes and objects in VBA has made it easy to create a login session which could be manipulated to show users information that is only relevant to them, while showing staff members a span of information from the whole database. On the other hand, an example of lacking security is the way passwords are stored – there is no in-built encryption and decryption for passwords in access, therefore I would have to use a complicated workaround such as coding an encryption and decryption function – this would make debug testing and password manipulation much more difficult, and I did not see the viability in doing so.

VBA is a rather simple language to learn, in part due to its close resemblance to English, but also due to it's huge online expert base. If I had a problem with a function or needed to know how to do something, the MSDN or expert forums would usually provide a clear and concise answer on how to solve the problem.

Overall, the language has been useful and effective in finding a solution. It has many in-built features which I have taken advantage of accordingly. I have managed to find workarounds for any complex functions that weren't in-built and have accordingly weighed up and implemented.

Comparing my system to similar commercially available systems.

When compared to commercially available systems, my system holds up very well. The solution I chose is part of a market with a smaller outreach, considering these systems are designed for musicians in the local community. However, this makes it easier for me to have an advantage over other systems, as I can take the functions they have and vastly improve on them.

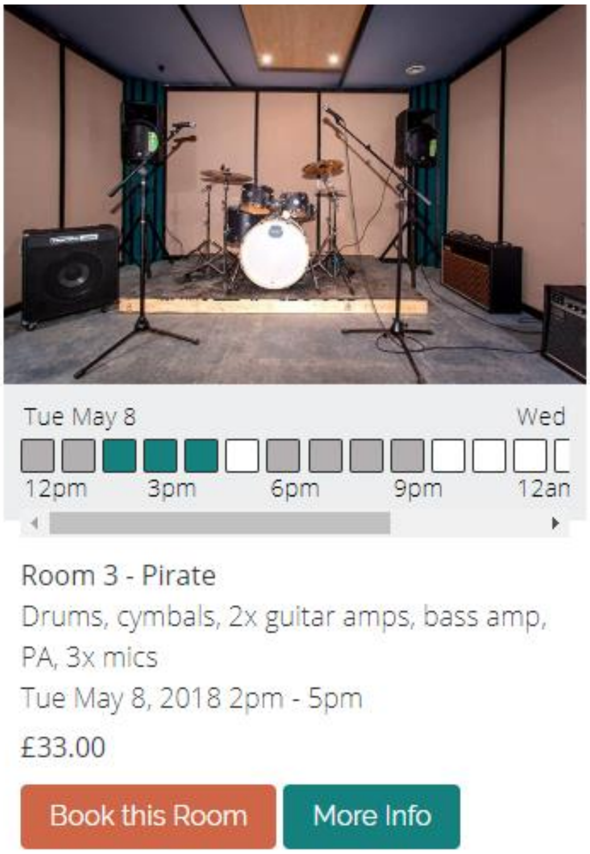
For additional clarity, I have created bullet points detailing the similarities of my chosen similar system (Pirate Studios) as this was the closest to my concept:

- Create bookings for music rehearsal rooms.
- Register an account for user identity & login to use the system.
- View bookings that have been made and get an output with relevant information. (invoice)
- Avoid double bookings.
- Accept payment and create booking instantly.

The majority of similar systems I researched lacked any automation. The customer would have to manually email or phone the company behind the room hire. I have been able to implement an automated system where the user can create a booking simply by inputting the relevant data and paying. The only other system I have found with similar automation is Pirate Studios. However, I believe my UI is much more user friendly and intuitive to Pirate Studios' UI. One thing that Pirate Studios does better is that they have information on peak times as well as images and virtual tours of their rooms. I could not implement peak times due to not having a previous archive of data, and cannot analyse data trends. With my time constraints and lack of staff identity, I could not take pictures of the rooms or find suitable alternatives.

One thing clearly superior about my system to the majority of other systems is the instant gratification. Users can use their computer and book a room within 10 minutes, whereas with other systems users will typically have to wait for a response from staff and then may even receive the bad news that the slot they want is already booked. In my system they would receive an error and instantly try new times or rooms. They also have the option to pay instantly, with discounts being applied if they qualify.

A downfall of my systems is that other systems may be more mobile friendly. Depending on how the system is hosted, mine may be incompatible with mobile devices. Every other similar system I have looked at is optimised for mobile in some way. This can be improved on by web hosting my program, or coding it in a different language where I can create an app or more mobile-friendly web version.



Room 3 - Pirate

Drums, cymbals, 2x guitar amps, bass amp, PA, 3x mics

Tue May 8, 2018 2pm - 5pm

£33.00

An example of pirates visual display of the room and availability as well as written info about the room.

Good features and shortcomings of my system, including potential improvements to be made.

My system has many advantageous features, which do exactly what was needed; however, there are also some pitfalls worth noting.

One good feature of my system is the booking manipulation and handling. Users can create and modify bookings as they please. Bookings can be paid for at the time of creation, and double bookings will be detected and avoided. I have implemented dynamic price calculations based on user requirements which provides a price instantly.

On the other hand, one feature my system is lacking is a user-friendly way to delete a booking. In order to cancel a booking, users must contact staff at Cathays through email or phone and have it manually corrected. If given more time, I could have implemented this feature by creating a deletion and cash refund system. Users also cannot delete previous bookings, and these bookings show up in their booking view. This may lead to clutter and confusion. To combat this, I could have separated bookings by dividing them into categories each week/month/year in the style of a grandfather-father-son system. I could then store archives in a place that's hidden to customers.

Another advantage of my system is its ease of use and user friendly interface. The UI consists of large buttons and labels, with everything clearly annotated to make it easy to understand how to perform each function. This will make the system extremely easy to understand and first glance and therefore any typical user will be able to use it without training.

One shortcoming to consider, however, is that individuals with a disability such as a vision impairment may struggle to use the system. While the interface objects are very large, which will satisfy most in this category, it might be helpful to implement a text to speech feature to describe the page and options.

A particularly useful feature of my system is the user activity log that staff have access to. This allows staff to trace any actions that have been taken on the system, therefore providing evidence in case of some sort of dispute such as a customer complaining that they made a booking but the room was unavailable when they arrived at the centre.

An improvement that could be made to this feature is the lack of management available. The user log is sorted so the most recent actions appear at the top, however it would be extremely efficient to have weekly backups in a type of grandfather-father-son system. These can remain on the system and can be sorted into categories that can be searched to improve speed of searching. Month old logs could be archived for historical purposes.

Significant potential improvements to be made to improve effectiveness of the system.

Firstly, when creating a booking the user must select the quantity of items they wish to borrow then manually type which items it is they want. Rather than doing this, it may be more intuitive to have staff members update what instruments are in stock and have this update the create booking form. This will prevent demands that are too extreme from being made and will reduce the amount of typing needed to make a booking.

Another potential improvement is the addition of a visual display of what rooms/times are already booked. This would allow users to see what is already booked at a glance and would therefore not have to continuously try different time slots. This would further speed up the booking process.

More extravagance could be added to my system with the implementation of more information about the rooms available to book as well as the equipment available to borrow. I could add dimensions of the rooms as well as multiple images and potentially a 3D virtual tour of the rooms, which wouldn't be too unrealistic to make using a program such as Google Sketchup. As for equipment lending, I could add a stock system that includes make and model of the instruments, their availability, an ability to rent them externally, a check-up of whether the instruments have been cleaned and serviced and more.

Lastly, a significant improvement in today's technological society would be to send the report as a text or email. The report currently only presents itself in print-preview format, however it may prove useful to the user if they had an option to receive an email or text as they would be able to save this to a mobile device and avoid printing altogether. Many people would prefer this more concise and eco-friendly option.

Evaluation of my own performance.

When looking at the system and comparing it with time constraints and speed of learning, I believe there are many points of discussion to note.

Firstly, I believe I have succeeded in my aim of creating an intuitive and user-friendly system that does what it sets out to do. The final product is aesthetically pleasing and easy to use, based on the feedback I've been given. It was important for me to keep the theme of the system consistent and easy-to-follow, and I believe I have accomplished this.

Another positive about this project is that it effectively follows the design. I managed to put in enough detail in terms of UI and functionality for me to use it as a guide on the creation of the system. This has made it so the system has full functionality and is as detailed as I set out for it to be.

Lastly, I believe that my time was well-managed. As I volunteered to teach Access and VBA to my A2 class without any external input, therefore I was set back 3 hours each Tuesday from the beginning of the year, meaning I have generally lost tens of hours compared to classmates. Yet when looking at my system and documentation, I am happy with the work that has been done in the time I've had to do it. This is inclusive of the time I've spent learning VBA independently to a proficiency level where I could teach it to my class.

In terms of negative aspects, I believe there may be a lack of extravagance in my system. In technology today, people want a single app to do a huge variety of functions; I could have made my system a hybrid between an instrument stock system, a booking system, a musician's forum and an advertising platform. Of course, with the limitations providing this may have not been possible, however I could have attempted to add extra features such as a notification system to the dashboard or a general announcements message board.

Furthermore, in hindsight I didn't collect enough detailed feedback from system users. If I proposed more detailed questions in any questionnaires or interviews I conducted, I could have potentially excelled in user friendliness as I would have an even better idea of what people specifically wanted to see.

Specific changes I would make to approach the problem.

While VBA was efficient in completing the task, it is not my preferred language and I was inevitably slower in that I had to learn it and use it. I believe if I was to do a similar project again, I would use web-based languages (i.e. HTML, PHP, JavaScript and CSS) with MySQL so that I have a lot more control, familiarity and ultimately have a lot more functions available to me. This way, I could have implemented the features that were simply too time consuming or impossible in VBA.

There are multiple improvements I could make if I were to do something like this. Multiple users could log in at the community centre and use the system without fear of corruption. The application would be mobile friendly. Applets and other widgets could be put onto the system to make it more intuitive. Encryption would be much easier to handle using MySQL paired with a database that I'd have more control over. Lastly, I would be more comfortable in this language and would produce work and documentation more efficiently and more quickly.

I will bullet point some more specific changes to make:

- Increased security by encrypting every piece of data in every table.
- More automated backup and archiving methods
- Potentially more secure login for staff members, such as needing a physical hardware ledger which generates a new password for them every ten minutes. This makes it extremely unlikely for hackers to gain access to administrator rights.
- Separate staff into different permissions groups, so that administrators have more access permissions than people who are new to the company.
- Add more bonuses for artists such as a loyalty program which stores points to allow them to receive discounts or spend the points on free sessions.
- Create a payment system which is up to the same standard as commercially available systems, including bank level encryption. I could potentially implement this by obtaining a license to use a pre-made API from official banking system programmers.
- Continue to work on and improve the system so that more and more features come in to keep up with the ever-changing technological world.
- Adding small, non-intrusive advertisements for other aspects of the community centre aside from music to allow the centre to perform more successfully as a whole.
- More measures taken to reduce the amount of errors that users will receive, for example, adding more clear labels telling users that they need to click a booking when generating an invoice.

Did I follow my objectives?

- 1 - Allow the user to add or modify bookings

I have completed this by allowing users to create bookings directly from their control panel. Users are also able to modify a booking just as easily. The system allows the user to pick a time, a date, equipment to lend and even a specific area of preference – what more could they ask for in a simple booking? The user is also able to pay for a booking, which is a crucial step in the process – this way they could theoretically use the system without even interacting with a staff member.

- 2 - Allow new users to register themselves

This is achieved in the register form the users can access straight from the login page. The form requires the users to input the most important information, yet they can choose to avoid inputting sensitive data such as card information if they wish. This being said, I'm happy with the state of registration in my system for new artists. The user is able to use the account they made as soon as they make it, which is also crucial to the user friendliness and speed efficiency of my system.

- 3 - Allow staff or users to log in

This has been demonstrated in the login screen that is visible to users as soon as they start the system. As long as staff or users have an account, they will be able to login. Staff and Artists can even log in from the same login screen and be brought to their own respective control panels – this removes the need for a convoluted split login system while keeping the system simple and easy to use. When logged in, the logged in username is saved, which creates a login session for the person who is logged in. This is just a further example of good functionality in this system, which comes in handy in many parts of the system.

- 4 - Allow staff to register new staff members

Similar to registering new users, staff are asked for the required information. The staff member is immediately able to use the system with elevated privileges. I'm happy with this intuitive and easy to use registration system that is quickly available from existing staff control panels.

- 5 - Allow staff to add or modify bookings for any user

Staff are able to add bookings and modify them just as users are, however staff have much more control over this. They are able to add bookings just the same, however when modifying bookings, they are given a list of all bookings. On one hand, this may pose a threat to security and may also seem cluttered and less user friendly. On the other hand, this can give staff the complete control they need to keep tabs on the company and be able to aid customers at a moment's notice, therefore I decided to keep all bookings being shown.

- 6 - Allow staff to create an invoice of any specific booking / Allow users to receive an invoice of their booking

Staff are able to view all bookings and create an invoice based on them, simply by clicking the booking in a list. I have also managed to implement this functionality on the user side, with the login session being used to filter out any bookings that weren't made by that user. This increases security in that customers cannot see other customers bookings. The invoices are easy to create, however at the moment they can only be printed. This could easily be improved by implementing an email system – however this would take more client-side setup.

- 7 - Prevent double bookings from happening

Double bookings are prevented in my system – when a user makes a booking, the system will check three things – the time, the date and the area. If any areas overlap with any times on a specific date, an error message will be shown to the user asking them to choose a different area or time. This will prevent the catastrophes of sending people away due to double booking and will further reduce staff intervention over disputes.

- 8 - Allow staff to modify users' account data

As staff have elevated permissions and can be trusted with more sensitive information, they can see and modify all of the users on the database. This would need to happen if a user contacts the support and asks for data to be changed – which is vital in keeping the system in check with the data protection act.

- 9, 10, 11, 12 & 13 - Be intuitive and efficient so it is fit for usage by anyone, prevent users from accessing other people's data, be much quicker to use than the paper system, be much safer and securer than the paper system & give users and staff a pleasant experience while dealing with the system.

These objectives are either focused on UI or security. As for UI, the system follows a very consistent theme of bright colours and obvious labels. Each function of the system is easy to use or even figure out if it isn't implied at first glance. The navigation is fluid and tabs are kept on every form that is opened so that the user doesn't get lost within the system.

As for security, measures have been taken in order to maximise the safety of data. Card validation checks are used for purchases. Password validation checks are used for registration. The login session restricts what customers can see on the system. All in all, the system is secure and the average computer user would be unable to compromise another customer's data.

Final Review

Overall, I am pleased with my system. While there are some small pitfalls, the system works as intended and as specified. Of course more features could be added, but when compared to other systems similar to this one, my system holds up as being a very strong candidate even against professional systems that are currently in use. The essence of a solution to this problem has been captured and practically applied to produce a functional solution which could hold up for decades if it were used in a professional environment. My usage of an agile approach in creation has helped me deliver this software at a deadline with little to no bugs – leaving a refined and professional system ready for commercial use.

If I was to conduct a similar solution development to a similar problem, I would potentially switch my coding language to a web based one rather than VBA. I would potentially put even more detail into designing the system so that absolutely nothing has to be done ‘on the fly’. I would brainstorm every viable feature and attempt to set aside more time for the project to be completed in. This way, I would have a more extensive and functional system which still retains the core good qualities of this one.