**Nails By Katie – Computer science A level project**

**Introduction:**

As Katie’s business expanded, it became apparent that there was a need to transition from a paper-based booking system to a computer-based booking system where Katie could view important information regarding her business, her clients, and her stock quickly and easily. This need has led to the creation of the Nails by Katie Booking System.

The system allows for a user to book, view, and amend an appointment and allows for an admin such as Katie to view stock levels, appointments and analytics such as most popular day of the week or most popular type of appointment.

Whilst there are other booking systems available such as Jobber, this system is specifically tailed to the customer and their experience allowing them to edit and view their appointments using a unique ID.

**Stakeholders:**

The stakeholders for this program would be anyone who is a client for Katie and her business. Other stakeholders would be Katie as the program also involves a high level of admin capabilities such as viewing analytics, vital to businesses.

**Questionnaire:**

To get a better understand about the need and wants of our client Katie and her clients to the business, two questionnaires were devised in order to understand their needs and wants. 1 questionnaire would be targeted at Katie, trying to understand her current method, how it helps her, how it doesn’t and what she thinks might be a better idea. The second questionnaire is intended to her clients, asking how swift they believe the current method is, how easy it is to edit, cancel and book appointments.

Question 1: What current method of booking do you use? And how do you communicate this with the client?

Justification: understanding the current method of communication and system is important to understand the problem the client has

Response: Paper Day diary, communicating appointments through face-to-face contact or text messages

Question 2: If I asked you to tell me the current stock level of a product, how easy and accurately could you answer me?

Justification: assess whether the client can already do things that the new program is meant to be able to do e.g., stock

Response: whilst I don’t keep too much stock, it would be difficult to say exactly how many different appointments I could carry out with my current stock. Despite this I can work out a rough idea by viewing the contents.

Question 3: What is your primary method of storing appointments? And depending on your answer, do you have copies of the documents?

Justification: this question was to compare the current method of storing, a paper system with no backups compared to a database which could have backups of it.

Response: Paper diary with no backups or duplicates

Question 4: How easy is it to see your most popular product or type of appointment?

Justification: to ascertain if my client’s current diary outperforms my program and its intended functions

Response: Look through my diary and take note of the most popular appointment but this would require a large amount of time as id have to go through each individual page and day and tally each appointment.

Question 5: What aspect(s) about your current booking system would you say works in your favour for your area of work?

Justification: understanding the advantages of their current method allows me to better tailor my own program to their needs and wats.

Response: This method allows for a large amount of flexibility and allows me to customise the time needed to carry out a client’s appointment depending on their needs.

Question 6: What aspect(s) about your current booking system would you say doesn’t work well in your favour for your area of work?

Justification: understanding the disadvantages of my client’s program allows me to make a special effort to improve these areas when developing my own program.

Response: My method can lead to human errors, appointment put in at the wrong time or wrong appointment type. I am also needed to input each appointment and takes a large amount of time to enter everyone’s appointment

Second Questionnaire:

Question 1: What is your primary method of communicating bookings, including edits and confirmations, with Katie?

Justification: understanding how the clients communicate their appointment is important to understand the issue that the client currently faces.

Response: Text messages, face-to-face or phone calls

Question 2: Regarding the method above, what would you say are the benefits and drawbacks of this method?

Justification: the answer of this question will better inform the development of the program as I would be able to maintain areas of the advantages whilst eliminating areas of disadvantage.

Response: Can discuss what needs to be booked in and can view available times. Takes time to get a reply from Katie, so appointments cannot be instantly booked.

Question 3: If the booking system changed to online using a new method, would you remain a client with Katie and her services?

Justification: migrating to a new system wouldn’t be beneficial if my client’s clients base weren’t prepared to migrate to a new system.

Response: Yes, the method of booking wouldn’t change my opinion on Katie and her services.

Question 4: if you could change one thing about the current method of booking system what would you change?

Justification: understanding where the client wants the system to improve is important when creating the program and considering its direction to best apply to the clients.

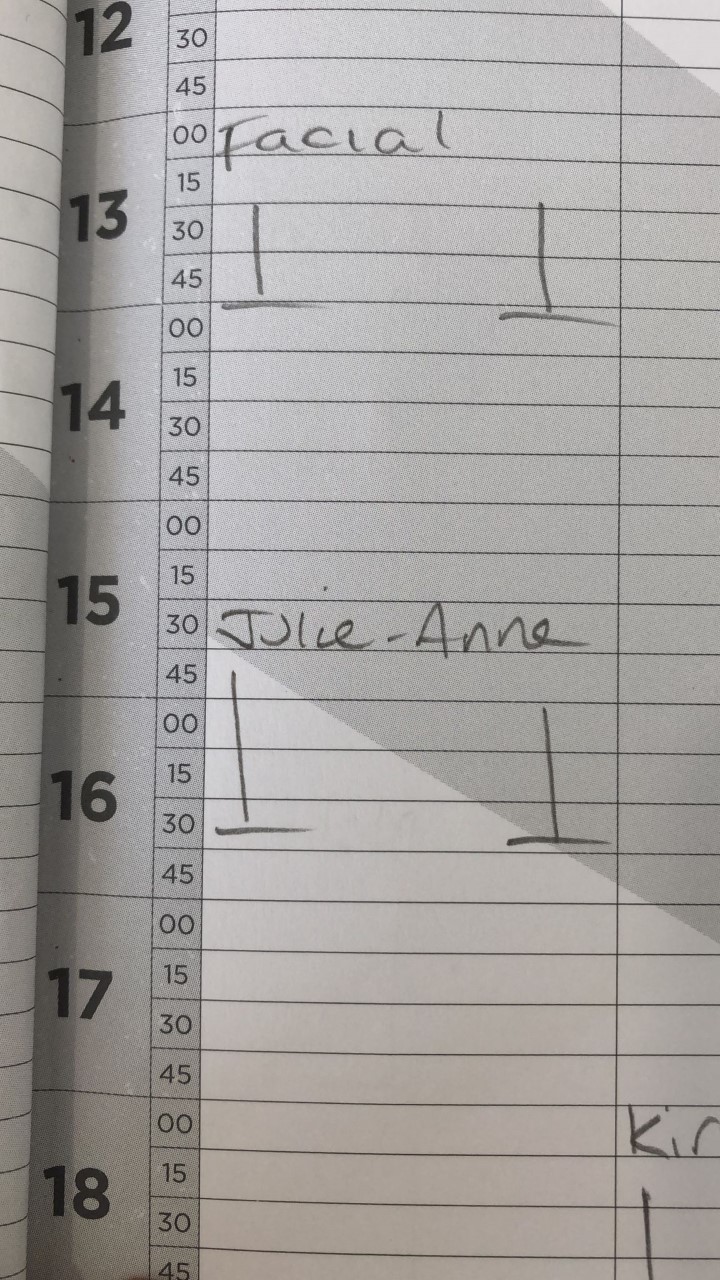
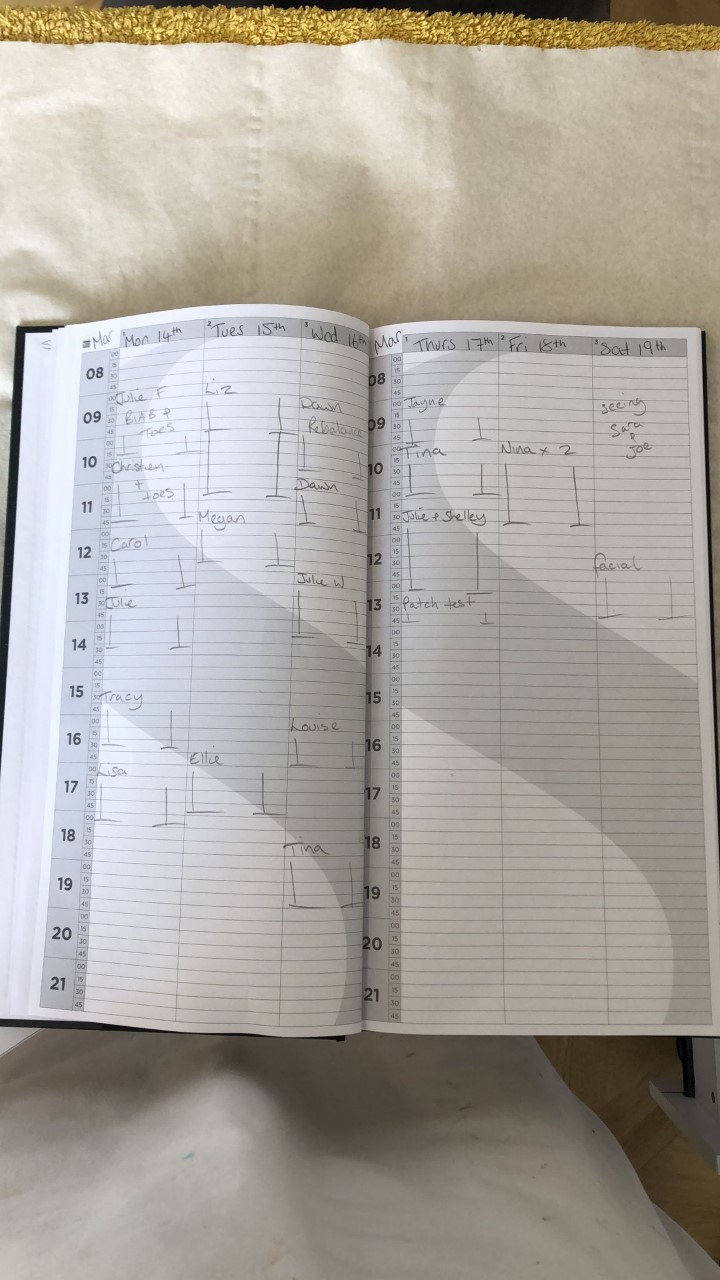
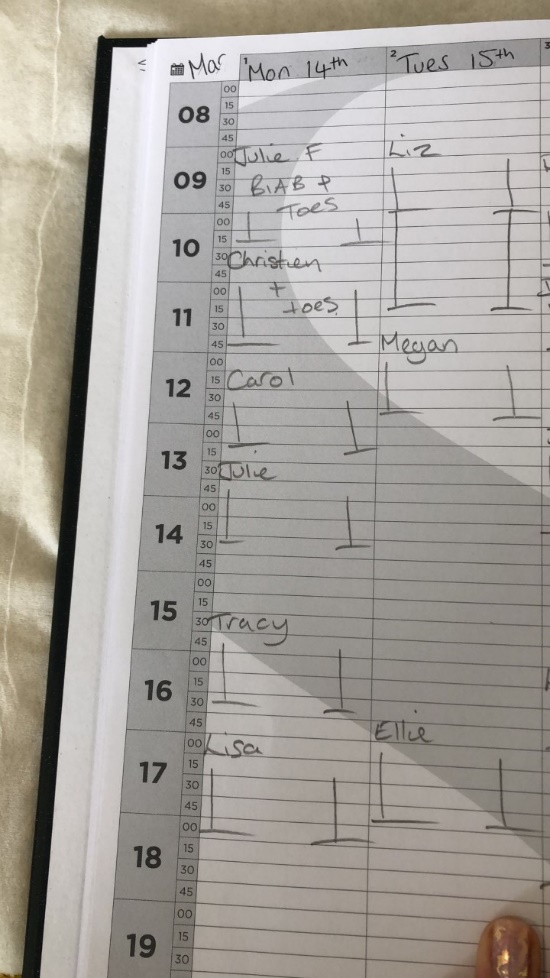
Response: A reminder to be sent out 48 hours before an appointment.

**Analysis**

The system developed is designed to help the user book appointments in an easy manner, as well as allow Katie to view important statistics about her business, which might influence her focus of the business. Katie currently uses a paper-based system which she writes the clients name, their appointment type and shows a visual representation of the time it’ll take to carry out their appointment. This method is good for Katie as it allows her to easily spot times in the day that are free, and which aren’t. This method also allows Katie to see who is booked on that day and what appointment they have booked. This method isn’t available to the client/end-user, so communication is required to confirm or edit their booking which requires larger amounts of time compared to querying a database of their appointments. Katie suggests key attributes of the booking system would be ease of booking (for Katie and client) as well as ease of viewing appointments (Katie).

The current paper-based system Katie currently employs:

The system involves printing the name of the client and then drawing the time it takes to complete the appointment. Each column of the diary represents a time which Katie writes the date on top of. This method is time consuming and prone to human error but allows for important notes or information to be added to the appointment. Another important drawback of this is the time it takes to query this collection of appointment. For example, if someone wanted to cancel their appointment, Katie would have to manually search the diary for their appointment and rub it out to cancel it. This is time consuming and prone to error.



**Solvability:**

The problem that the client current faces is the time it takes to book and edit multiple appointments whilst be able to keep track of important information which will help her tailor her business to the needs and wants of the consumer. This issue is solvable using a computational method. The current method is recording each appointment by hand and storing this in a diary. But this method is flawed by the time it takes as well as the possibility for the loss of data if the diary was damaged or lost. This method doesn’t allow for tracking of products or clients.

The problem is computable because it can be solved using algorithms which tackle each aspect of the client’s problem e.g., booking an appointment (the algorithm allows the user to input various amounts of data into entry boxes and creates an appointment and unique ID for them).

By using a computer, the solution to the problem can be achieved within a finite amount of time and the data and program can be stored in very small capacity (approximately 10kb for the database and 50kb for the program) therefore requiring less storage and hardware power.

The output of the computer uses abstraction to simplify the entry for the user. The user enters information into labelled entry boxes rather than straight into the database in a random order. The display of the information is also shown in the program in a format which the user can understand rather than viewing it straight from the database. Using a database is important as it can be stored on a server which multiple devices would be able to access and book appointments to them.

Validation is carried out by the program so the user doesn’t need to validate their own information against the parameter for each field or that their appointment doesn’t contradict another appointment.

**Problem analysis and decomposition:**

The problem can be broken down into different paths which the program will take a user through.

1. User interface:
   1. Creating a menu which separates a client and an admin
   2. Client side has 3 options; book, view or edit appointment
   3. Booking an appointment allows the user to input required information into entry boxes and then enters this information to the database
   4. Viewing an appointment allows the user to input an ID or a surname and displays a specific or a range of appointments based off the information the user entered
   5. Editing an appointment requires an ID and from that the user can select a field to change and then the program makes this change to the database
   6. The admin page can either view stock, plot graphs (analytics) or view appointments
   7. The view appointments page can chose between filtering by day, name or ID. The entry box will expect different inputs based on this choice and displays information based off the input
   8. Viewing, editing or adding stock consists of typing in a product and viewing the products level of stock. This window can also lead to adding a new product along with a current stock level. Editing a stock level involves typing in a product and typing in a new stock level.
   9. Analytics will be in the form of graphs by choosing a field such as products or clients and then the graph shows the quantity of this product or person that has been used or made an appointment.

**Graphical user interface

Description automatically generated with low confidenceStructure diagram:**

1. Data representation:
   1. It Is important to consider how the information will be saved and accessed when the user makes an enquiry. The data consists of strings and numbers which allows the data to be stored using a csv. But a major drawback of this is that individual record cannot be changed meaning the entire database has to be re-written when a small change is made, making it a poor solution to large data sets. But on a small scale such as products, this is an applicable method.

**Hardware:**

A computer running this program will need a mouse, keyboard, and a screen to display the program on. The program will need sufficient storage and memory space to run the program, but this amount is so small compared to current specifications of modern computers. The power needed from a CPU is relatively low as there aren’t complex calculations or rendering so any CPU should be suitable.

using the take manager, I can see that whilst being used the program uses 15.5MB of memory and 0.4% of the CPU (4 core 8 thread CPU at 1.5ghz). this test shows that the program doesn’t require high specification computers or large amounts of memory.

The alternative for a keyboard and mouse would be a touchscreen where you can type your entries or press on the screen to press buttons. Although most computers and laptops don’t support touchscreen, although this would be essential if the app was brought onto a mobile phone such as tablet or smartphone.

**Software:**

The software that I’m using to create this program is python on windows 10. This is because I have the most experience with python and its libraries such as tkinter or date and time library. Python is also applicable to other operating systems such as Linux or macOS, meaning the program can be portable between devices on different operating systems.

Tkinter was the choice of graphical user interface for the project because this comes standard with python, making it more applicable to un-experienced users. Another library used is the dateandtime library which allowed me to compare dates within the program. This library also comes as standard to the user. A final library that I would user would be ‘matplotlib’, a graph plotting library that I would use to plot bar graphs to show information, matplotlib does not come as standard but as this feature is only needed for an admin it does not require for every user to install this. There is another graph plotting library called sympy but this library specialises in plotting mathematical equations rather than predetermined data.

**Research:**

Currently in the market, there are other booking methods and analytical analysis programs that the client could use, however these methods are mostly web based.

One example would be Jobber.

Graphical user interface, text, application, chat or text message

Description automatically generatedJobber is a paid for platform that allows a user to keep track of clients and their appointments. When signing up it asks a few questions about the industry, the business and contact details. The photo to the right shows a client profile menu which allows a user to input basic information about them and can be used when booking an appointment about them.

An admin can create an appointment on an available day, linking a client’s profile to an appointment on a specific day with a variable time. This appointment can also be set up to be re-occurring appointment.

**Calendar

Description automatically generated**Jobber has special features which make payments easier, as well as having reports which can track over 20 metrics corresponding to the business and their activities.

Graphical user interface, application

Description automatically generatedAn area where Jobber does not apply well to the client is the lack of stock tracking. When setting up an account with Jobber it asks for an industry which, I assume, would tailor the experience to that industry, but as my client’s industry did not show up, there is no tailoring. This is where my program better applies to my client as it is very specific to their business model, as well as the app is accessible to their clients, removing the necessary admin inputs that Jobber requires. Although Jobber does allow for mobile phone access, allowing convenient booking of appointments and viewing of other aspects.

Jobber has an easy to view menu with a list of functions running down the left-hand side, although the headings are a little vague as they aren’t specific to a business (a consequence of making something to apply to different business’). The booking aspect does make it very clear to the user what times are taken or free for the user to book on top of.

Graphical user interface, text, application

Description automatically generated**Comparison between Jobber and my own program:**

In comparison to Jobber, my program is specifically tailored to my client’s industry and business, allowing for stock levels and specific job types.

A picture containing text

Description automatically generated

**Graphical user interface, text, chat or text message

Description automatically generated**Another important comparison is how a client can access their appointment and create booking using my application whereas using jobber, it is only an admin that can enter, view, and edit information. This is important because as my client’s business is growing, they require a system that a client can use to create and manage their own appointments without the need for an admin.

Jobber does hold more information about a client including contact information, or a photo to show what they look like. This is something that my program could develop into, having a database which has all the clients including date of birth or contact details and then instead of inputting their name every time they want to book an appointment, they can just reference themselves from the database.

My program allows for easier traversal for both users as it is specifically tailored to a booking system for a service, specifically nails. This allows for specific headings of windows making it easy for the user to traverse.

**Reference to website use**

Currently the program is only on a computer, using software that the user would launch rather than a website. The first stage would be able to create a website with the same functionality and usability as my program, but with it being on a website, you can have a database hosted on a server which can collect information about different appointments and other metrics such as stock. Having this on a website would allow anyone to book an appointment from their phone, tablet, or computer without having to download the software to run. On a website also allows the database to remain consistent and not having out of date copies on different people’s systems. Linking back to an improvement a client suggested from the questionnaire would be a reminder via email or text message which reminds the client of their appointment.

**Justification:**

One aspect of the program that needs justifying is why the program uses entry boxes over drop-down menus when a user views or edits their appointment. Currently they type in their unique ID which then queries the database for their appointment, but an alternative method could be to have a drop-down menu which shows all the possible unique ID’s. this isn’t a method used because there is the possibility to have 1000s of appointments saved to the database and wouldn’t be feasible to display all these appointments

Another area to justify is why the program doesn’t suggest times e.g., 1 week or 1 month from the current date is that each client takes appointment at different times and has different amounts of time in between each appointment. This factor is different for each client meaning there wouldn’t have a large impact if this was implicated.

To justify using bar graphs to show different information to an admin the information being displayed needs to be understood. The information on each axis will be a quantity on the y-axis and either a person, product, or appointment type on the x-axis. This applies to a bar graph as it’s easy to understand this information using this method and showing a visual representation that the client knows that the higher the better, makes it easy to understand quickly.

**Success Criteria:**

|  |  |  |
| --- | --- | --- |
|  | Success Criteria | Explanation |
| 1. | Creating a main menu which the user can easily operate and traverse | A main menu which an end-user can easily understand and explore through labelled buttons. The menu should also be helpful to the user in helping them find the specific part of the program they require. This can be done by clear labels and logical paths. |
| 2. | Easily accessible booking menu | Booking menu is easy to access and input information into. Entering a name including a forename and surname, and a date. Then the user can select times and appointment types via drop-down menus. The booking process should only accept valid information and explain to the user why it isn’t valid (improving usability) and only accept valid date and times which don’t overlap with other appointments. Different dialogues will be shown to the user depending on the error such as ‘invalid date’ or ‘date and time taken’. |
| 3. | Clear viewing of the user’s appointment | The user’s information should be easily presented so it is easy to read and understand. The layout of information should be in a clear and logical path displayed on the screen, such as the layout of booking an appointment. The information should be easily accessible to the user through their surname or ID. Searching through a surname will display more than one appointment but will only show valid appointments. When the search is carried out, the ID is checked to be valid, and the program shows an error if its invalid or there is not corresponding appointment. |
| 4. | Easy to edit appointments using a unique ID | The user should be able to edit their appointment by entering a unique ID, selecting the field to change, entering the change, and saving it. The program will then display that the save was successful or display a corresponding error. The user should then be able to view their updated appointment after the updated has been saved without closing the program. The ID that the user enters is validated when they attempt to edit that appointment. The information about the appointment is also validated. For example, if they wish to change the date, it would be validated against other appointment to ensure that the appointments aren’t booked on top of the same time slot. |
| 5. | Simple for the user to cancel appointment | The user has the option to cancel their appointment when editing it. This action is done by entering their unique ID to an entry box, then clicking cancel which will open a confirmation window to confirm the cancel. This should free up their date and time slot for another user and their unique Id should become invalid so they can’t view an invalid appointment. The ID is validated when the user clicks the cancel button and displays an error message based on if the ID was invalid or the appointment doesn’t exist. |
| 6. | Admin should be able to view specific appointment of each client as well as see a range of appointments filtered by day or type of appointment. | The admin can view specific appointments by searching a surname or unique ID which are validated to ensure they exist and follow the validation rules. The admin should also be able to view all appointments (in order) of an upcoming day by entering the date they wish to view (date would be validated, ensuring that there are valid appointments on that day). This result could be printed out from a printer (requires a printer and an internet connection) , so the admin has a paper copy of that day’s appointments. Filtering by type will allow Katie to prepare for that appointment type. |
| 7. | Admin can view stock levels of all current recorded products and can easily edit them | The admin should be able to view specific products by searching the product name through an entry box or drop-down list. The entered product is then validated against saved products and returns an error if no stock item is found. The admin should also be able to add, edit or delete new products through a different window. Stock levels should be able to be altered by the admin when products are used or more stock comes in. Buttons would be used to increment or decrease stock levels, or an entry box to increase by a specific amount e.g., 50. A save button is to be used when adding, deleting, or altering stock levels. This would carry out validation of new information and asking for confirmation that the change is correct. |
| 8. | Admin should be able to view certain analytics based on the business | The program should plot graphs which show the most popular products/appointment types, most popular days, or time and the highest re-occurring clients which has the most appointments booked with Katie. The window should have a drop-down list to alter the field to plot e.g., appointment type or popular product. A ‘plot’ button would be used to initiate the process of validating the field that the admin entered the entry box. For example, the graph wouldn’t plot if the date entered was invalid. The graph should be easy for the admin to plot and understand using clear axis and labelling. The graphs are to be bar graphs as these are easy to understand and applicable to the data to be displayed. X axis would be product, date, client etc and y axis would be quantity. |

**Limitations:**

All solutions to the problem aren’t feasible to be explored due to insufficient time as well as developing my understanding of Python.

Lack of connection to the internet:

This lack of connection means that the user cannot keep a consistent database across multiple platforms and would have to manually copy the database across devices. Referring to a website version, this would require a connection to the internet to update and query the database of information that a client or admin would use. A connection to the internet allows for a reminder via email or text to be sent to a client, linking back to the questionnaire and a suggested improvement.

Lack of privacy:

The program will contain information that is private about the business and its clients. Currently the admin page does not require any username or password to view and edit sensitive information. This means that any user could view the sensitive information. To prevent this a system should be implemented to ask for a username and password and validate this against the correct password.

Another area of a lack of privacy would be viewing and editing appointments. This is because, currently unique ID, whilst unique per appointment, aren’t very random (increasing by 1 per appointment) and these IDs could easily be guessed by someone that doesn’t know someone else’s unique ID. A fix for this would be to extend the length of the unique ID and having a mix of letters and numbers to reduce the chance of someone being able to guess an ID. A final point is that anyone can search a surname and find corresponding appointment. A fix for both these issues would be to create a log in portal requiring a username and password to book, view and edit appointments. This would use a table of clients to book appointments to.

Lack of customisation:

An area where the system isn’t applicable to the customers industry is the flexibility of making your own appointments, adjusting people’s appointment times. For example, my client might know that client no. 1 takes 1 hour to carry out the service, but client no. 2 takes 2 hours. This is a problem because there currently isn’t the ability to use custom times, whereas my client’s current method allows for much more flexibility, as mentioned in the survey.

Single user:

Currently the program is only on one central computer. The program would take the direction of a website to be used by multiple people across multiple devices. This is a limitation because currently all appointments require input from my client, and my program still requires this as a client cannot book, view etc from their own device and would have to use my client’s device.

No customer contact details:

Currently there is no connection between my client and their clients. As spoke about previously, my program doesn’t currently have a table that holds a clients contact details meaning if Katie needed to contact them, not knowing them personally, she wouldn’t be able to. This contrasts with Jobber and their client information. But this is an area that the program would branch into and having a table with multiple fields of data corresponding to a client, making it easier to book appointments without having to input their data every time. This table would also be important if there was a login portal to the client side of the program.

Another aspect of this would be to have a barcode or QR code which links to their personal details and accounts. This code could be used to login automatically without the need for a password or unique ID corresponding to that client. However this method would need a camera or barcode scanner in order to read the code which might not be applicable to most users.

Only use a keyboard and mouse:

Currently the only method of input would be to type the information into various entry boxes or use a mouse to select information from drop down menus. This method might not be applicable to different users and would prefer to use a microphone to book, view and edit their appointments by using voice recognition and speaking out their information. Some users may not be able to view the information presents to them on the screen so speaking out, using a computer speaker, the different labels to allow users to hear what information is required.