

**Name:** Stergious Aji

**Candidate Number:** 4211

**Centre Number:** 71887

# JD DOG CARE



**Lumen Christ College**

Software System Development



# Analysis

## Background

JD Dog Care is a successful business based at a farm location just outside of Newtown. The farm and the dog care business are owned by John and his wife Diane and has been in business for nearly seven years. The business has extensively grown and diversified over that time and with that came an overwhelming number of customers and demand. Diane successfully achieved an accredited professional dog grooming Diploma as well as a Certificate in Animal Care and Welfare. It is evident that Diane is well qualified in the care for animals, but her knowledge of business procedures is limited.

### Dog Grooming

JD Dog Care has offered a basic dog grooming service since when the opened. This involved booking dogs for scheduled appointments from Monday to Friday, 9am to 5pm. Diane records appointments in a diary and basic details are noted beside each entry, like the client's name, address and telephone number and their dog's name and the service option selected for the appointment. Diane insists on an additional 15 minutes for first time clients to become familiar with the dog and to discuss treatments and additional information about the dog. This adds an extra £5 fee to the final bill.

The service options include:

- Option 1 – Wash, shampoo and brush
- Option 2 – Wash, shampoo, brush and trim
- Option 3 – Wash, shampoo, brush and full cut

(Nails, teeth and ears can be incorporated with any of the service options for an extra fee)

One-off clients pay after the grooming service is completed while regular clients can choose to pay monthly preferably by standing order or by cash or card (Cheques are not accepted). Receipts are given to all clients after the service option has been paid for and monthly invoices are prepared for those who pay in a monthly contract. Clients who book three appointments in advance are given a 5% discount. John and Diane have recently introduced a range of additional treatments including allergy therapies. It was decided that these treatments require an initial assessment appointment and at least three more follow-ups over a period of six weeks. Clients must pay in advance for the assessment appointments in order to plan and set up the follow-up appointments. Invoices are sent before the completion of the three follow-up appointments.

The work at JD Dog Care is divided as follows: John and Diane take charge of the day to day running of the business like the maintenance and administration of the centre. They have recruited two full-time members of staff, Elaine and Richard, and one part-time staff member, Jane, who has committed to five half-day sessions per week. The staff members work in shifts and are responsible for caring for and grooming a dog for the allocated booking. At present, there are three grooming rooms at the centre meaning a maximum of three dogs can be booked at the one time during busy times.

## Business Problems

Evidently, JD Dog Care runs in a very traditional manner with all bookings, invoices and client and dog records being physically written on paper and stored in filing cabinets at the centre. This can be very taxing and time-consuming and is a huge risk as for one, if access to these important documents is not properly restricted, a security breach is more likely to occur which can have a disastrous impact on the business. Secondly, during busy times, not everything will be recorded which causes issues with appointments. The time allocated for each grooming option varies. This can be a problem and clients are often kept waiting before or after their appointments, as in many cases the appointments do not start or finish at the scheduled time.

Another huge problem is keeping track of invoices if they are paid or if they are not. Copies of these invoices are retained in the salon and should be marked 'Paid' after payment is received. This does not always happen, and clients have complained that they have been sent further invoices for services that have already been paid. Clients are normally given two weeks to pay before a reminder is sent out. If invoices remain unpaid, clients are issued a 'Reminder' and a duplicate invoice within three weeks of the original invoice. Further appointments should not be made for clients who have not paid until the bill is settled. It is extremely difficult to manage these accounts and unfortunately reminders' have been sent out incorrectly and further appointments have also been made.

All documents need to be very well organised or else it will lead to confusion and inconsistent data. At times the salon could be so busy that there is barely time to answer the phone. Clients sometimes leave a message on the answer machine to book an appointment. They often assume that Diane recognises the name of the dog and the owner and do not provide any further details. Of course, this is not the case as many dogs have the same name and not all owner details are recalled. Appointments may not be recorded as a result. Diane is always annoyed at having to repeatedly write the same details for her regular clients. Booking arrangements are clearly problematic, and John and Diane realise that they are not using the available resources effectively.

## Proposed Solution

In response to these issues, John and Diane have decided to redesign and improve the administration system at JD Dog Care by developing an IT system in hopes to automate parts of the process that are monotonous and time-consuming. This will increase efficiency of the staff members as their focus will be less on the administrative processes and more on the care of the animals in the centre, improving the quality of business. Security will be less of an issue with the software system as usernames and passwords can be implemented to restrict access to only qualified members of staff. The new system aims to make invoices for each of the client easy to keep track of and prevent further appointments from clients who have invoices yet to be paid. Reminders can automatically be sent through e-mail or post after the two-week and three-week periods. The system will ensure that all data is entered and inconsistencies in data storage will be minimised or even eliminated as data only must be entered once making booking appointments faster and more reliable.

## Methodologies Overview

### Standard Waterfall:

The waterfall model follows a simple, sequential development process where each phase depends on the deliverables of the previous one. This traditional approach requires detailed and precise up-front planning and documentation before any work is started. This means that all the user's requirements must be analysed and validated until there is a straightforward, clear path and end-goal to the project. The name 'waterfall' stems from how the development process mainly flows in one direction, downwards, like a waterfall, through the phases, conception, to analysis, design, implementation, testing and finally maintenance. Each stage of development proceeds in a strict order.

### Advantages:

Due to the waterfall model involving a lot of documentation defining each phase in the project, new developers can be quickly brought up to speed at any stage during the development process. The strict, rigid procedures also enforce discipline among the team so that their sole focus is on the current task. This makes large teams easier to manage for project managers as each stage has a specific deliverables and milestones that need to be kept on top of. In addition, during the analysis, the costs can be estimated with a fairly high degree of accuracy and so the budget could be changed or planned for at an earlier stage.

### Disadvantages:

Although the waterfall model is very robust, it has several disadvantages, for instance, it cannot accommodate changing environments and requirements. This method is only suitable for projects where the requirements are fixed and well-known. Another major drawback of the waterfall approach is the limited involvement of the end-user or the customer which increases the risk of customer dissatisfaction of the end product as they cannot give feedback during the design and implementation stages which may lead to redevelopment increasing the cost and time. Another critic is that the project management is very repressing, and this does not help encourage creativity within the team and use their skills in the best way. As the package is only integrated at the end of the lifecycle, often called a 'big bang', issues are likely to surface with incompatible code between developers due to miscommunication, resulting in redevelopment increasing time and money.



Figure 1 – Standard Waterfall Model

[https://www.pmi.org/kasimage/c4262e89-91f6-4b1d-a758-df86a2b484b5/p1\\_01.jpg](https://www.pmi.org/kasimage/c4262e89-91f6-4b1d-a758-df86a2b484b5/p1_01.jpg)

## Rapid Application Development (RAD):

Rapid Application Development is a type of incremental and iterative development model which was designed in response to the critiques of the waterfall model. RAD compresses several stages of the waterfall model into an iterative process that prioritises prototyping and a higher level of user involvement and feedback. The development team delivers a fully functional prototype of the product which is then met with an evaluation from the user or customer. The prototype is then refined to the user's needs and this process repeats until the prototype evolves into the final working product that will satisfy the user/customer. This is known as evolutionary prototyping, however, RAD also uses throw-away prototyping where instead of refining the existing prototype, a new version of the software package is created from the ground up accommodating the new needs of the user.

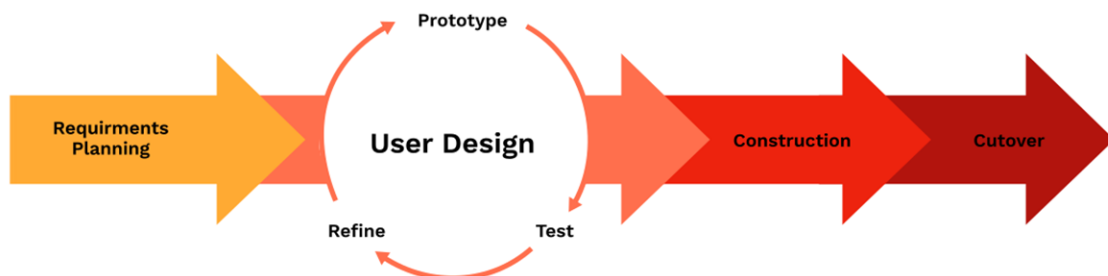
### Advantages:

One major advantage to RAD is that there is less up-front planning meaning more time can be spent on developing and testing the software product. Additionally, the user requirements do not need to be well defined in the beginning as prototyping takes care of this. RAD divides the project into several compartments for which separate prototypes can be developed in parallel, speeding up the development process and shortening the life-cycle. Changing environments and requirements can be easily incorporated in the evolving prototypes and so this reduces the risk of not meeting the customer's requests and therefore leads to a higher customer satisfaction.

### Disadvantages:

RAD is inapplicable to projects with low budgets as the cost of Computer-Aided Software Engineering (CASE) tools for modelling and automated code generation are very high. Additionally, RAD requires highly skilled developers and designers who have lots of experience and a large skill set in order to swiftly respond to the customer's changing requests. The RAD model also demands more frequent user involvement throughout the development lifecycle which would require a lot of commitment from the customer to achieve the targeted prototype in the given time frame.

Figure 2 – RAD Model



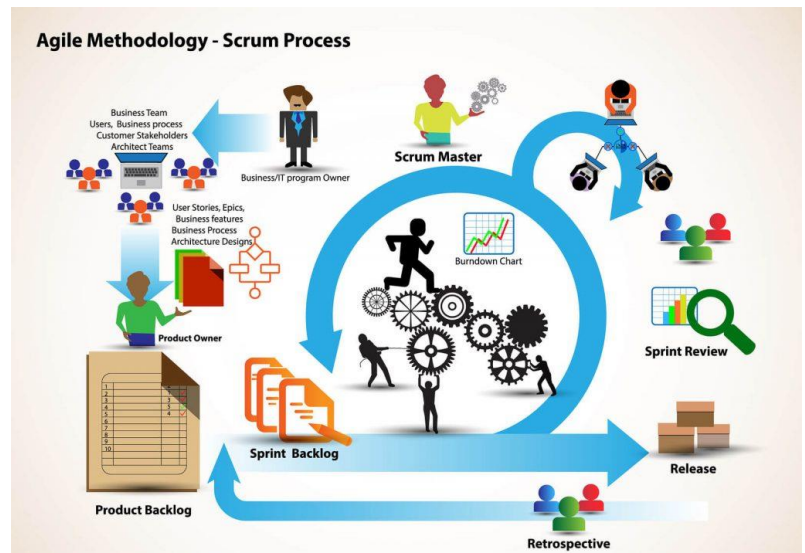
<https://roadmunk.com/guides/content/images/2019/07/method-ch1-6a.png>

Figure 3 – SCRUM Process

## SCRUM:

Scrum is a lightweight process framework and a subset of Agile methodologies. Scrum uses iterative and incremental practices with a main goal of increasing productivity and reducing the time taken to complete the project. It is designed for teams of up to nine to twelve developers who break their work into milestones that are completed in time-boxed iterations called sprints. These sprints last no more than two to four weeks and at the end of each sprint, a fully integrated and tested product that is potentially releasable must be released

for the customer to review. At the start of each day during a sprint, a Daily Scrum (or a Stand-Up meeting) is held at the same time and same place and is restricted to 15 minutes. During the Daily Scrum, each Team member explains what they completed yesterday, what they plan to complete today and if they have any issues that will stop them from completing their tasks.



<https://d30s2hykpf82zu.cloudfront.net/wp-content/uploads/2019/09/Scrum-Method-1024x751.jpg>

## Advantages:

Scrum is very efficient at enabling development teams to complete project deliverables in a rapid fashion and this is one of the main reasons that Scrum is the most popular Agile methodology, according to the 12<sup>th</sup> annual State of Agile report, 70% of software teams use Scrum or a Scrum hybrid. Scrum is extremely suitable for large projects as it breaks it up into manageable sprints for the development team. The Team in Scrum are self-organised and choose amongst themselves which parts each member will tackle in alignment with their specific skill-set. This improves efficiency and ultimately creates a software package that is better in quality. User feedback is an important aspect of Scrum, being an Agile model, and so this ensures that the final shippable product is in line with the user's standard and satisfaction.

## Disadvantages:

For Scrum to work at its best, all individuals must be fully dedicated and cooperative, including the customer and/or user. A consistent high level of communication between all participants in each Daily Scrum and meetings with stakeholders is essential for this framework to work effectively. If this is not the case, the project is bound to fail. In addition, Scrum requires highly experienced software developers and designers to reduce the risk of scope creep, which is when the project's scope grows uncontrollably at a point after the process has begun. Poor definition of tasks and requirements can also lead to failure as it leads to inaccuracies during the development process.



## Dynamic Systems Development Method (DSDM):

Dynamic Systems Development Method is another agile project delivery framework that focuses on the full project lifecycle, using an iterative and incremental approach with continuous user/customer involvement. It makes steps to ensure the feasibility and business sense of a project before it is created. DSDM emphasises on in-budget and on time product delivery with variable functionality to address projects of all sizes. In order to keep to the budget and be on time, DSDM uses MoSCoW prioritisation to leave out unnecessary requirements, reducing the project scope. The MoSCoW mnemonic simply means Must haves, Should haves, Could haves and Won't haves. DSDM makes a heavy use of prototyping and this ensures that all interested parties have a clear picture of the various aspects of the system.

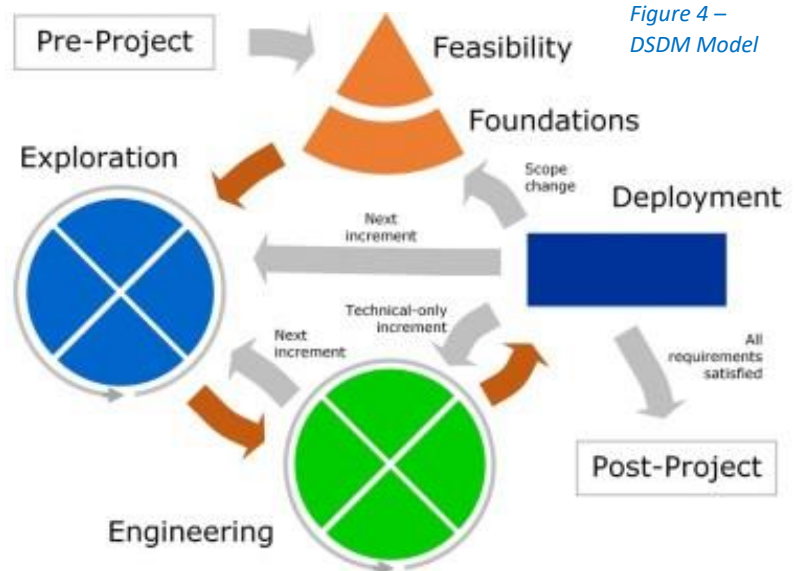


Figure 4 –  
DSDM Model

### Advantages:

In DSDM the basic functionality is delivered quickly with more functionality being delivered at regular time-boxed intervals in the form of the evolving prototype, this means the customer can see evidence of progress being made from very early on in the process. Like all agile methodologies, the constant user feedback helps ensure that the system being developed meets their needs requirements. As DSDM, unlike other methodologies, focuses on fixed time and resources, the system is always delivered on time and budget making it very reliable for the customer to invest in. Additionally, due to the early integration and testing in each increment, also known as Integrated Testing, hidden flaws will surface early in the process reducing the risk of redevelopment. MoSCoW prioritisation helps to eliminate requirements that are deemed unnecessary and would only waste time and money.

### Disadvantages:

DSDM focuses more on delivering the product within the proposed timescale, this can often lead to decreases in code robustness which could further lead to integrating issues later. Like other agile frameworks, DSDM requires full commitment from all individuals, even the user/customer, involved in the process for smooth and successful development. In addition, highly-skilled teams are essential in both the business and technical areas to ensure efficient workflow as DSDM is often a complex development process.



[https://upload.wikimedia.org/wikipedia/commons/thumb/8/84/Extreme\\_Programming.svg/1200px-Extreme\\_Programming.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/8/84/Extreme_Programming.svg/1200px-Extreme_Programming.svg.png)

## Extreme Programming (XP):

Extreme Programming is another subclass of the agile development framework and is specifically designed for software development projects. XP is distinguished for its immense focus on adaptability rather than on predictability. This is because of fluid nature of software development in general where requirements cannot be fully predicted from the beginning and will constantly changing as the project moves on. XP consists of very little to no documentation during the process and instead XP programmers start to generate code from the very beginning while listening to the customer. As a result, XP developers need to be naturally good listeners to enable them to understand and develop solutions which match the customers' needs and desires as close as possible. XP emphasises on the testing the function of the system and so tests are pre-written and then the application code is development to pass those initial challenges of the test, further tests are also written to try and break the code.



Figure 5 – XP Model

### Advantages:

XP is extremely fast-paced and so instead of taking a couple of years to complete a project, like in waterfall, XP projects only last for several months. This is because there is continuous integration and testing rather than a 'big bang' in the waterfall approach. Additionally, there is no wasting time writing lots of documentation that are often never used or seen again, further reducing the development lifecycle. This also benefits the developers as they will be much happier writing code rather than writing documentation leading to them being more creative during the design and development process. In XP, changing requirements are swiftly implemented throughout the development period, this reduces costs as it reduces the risk of a complete redevelopment.

### Disadvantages:

XP consists of a very fast-paced work environment and so demands for a highly motivated, stable and experienced development team who can rapidly translate the customers' words into a working software package. XP takes customer/user involvement to the extreme as they must always be on-site to answer questions and ensure that the software being developed is in line with their wishes. Due to little documentation in XP, constant changes are not kept track of, causing unexpected failures to occur and even when bugs are resolved, without accurate records, same errors may recur. XP can also be very stressful as developers have to keep to tight deadlines. This stress may cause them to make more mistakes, diminishing the quality of the final product leaving an unsatisfied user/customer.

## Chosen Methodology

I have chosen to pursue the traditional waterfall approach for my project, as I feel it is the most suitable for designing and developing a part of the new management system for JD Dog Care. The first reason being that I will be pursuing this task alone with minor collaboration with others and so this process eliminates most of the agile methodologies as they encourage a fully dedicated team of highly skilled developers to divide and complete the workload, leaving only the waterfall approach to be the most suitable. JD Dog Care already has an existing manual system in place and so the Waterfall methodology is the most appropriate to deal with implementing a manual system into software system with already existing and fully defined requirements for their system. The Waterfall model is easy to understand and follow for developers with minimal experience in the world of work, such as myself, and so I will be more proficient with the easy to follow, step by step sequential flow of the waterfall method. JD Dog Care has clear, well defined and mostly unchanging requirements for its software system and reliability in the system is somewhat critical for John and Diane to ensure smooth operations in their ever-growing business and hence, the Waterfall Methodology ensures that a successful software package of the highest quality will be delivered on time for the business owners.

## Chosen Project Area

Taking the limited resources and time constraints into account, I plan to focus on one area of the JD Dog Care Administration System to ensure the final product is of the highest quality. I have decided to tackle the Dog Grooming Services that JD Dog Care provides as this serves as the main foundation that the system can be built and expanded upon. The Dog Grooming Service was one of the first things that John and Diane have been and are still providing since the opening of the business and is one of the main things that put them on the map. For this reason, I believe that developing a software system to aid with this part of the business is vital to keep the business running and growing. As such, I will be developing an easy to use system to register and update clients and their dogs, as well as the staff members to allow for the business to grow. The system must also include ways to create and update bookings for grooming appointments and calculate relevant payments due.

## Gantt Charts



*Design*

## User Requirements

### 1. General Requirements

- 1.01. The software system must be user-friendly and easy to navigate around for new staff members pick up quickly.
- 1.02. All text in the application should be the correct size and font and use simple language to be easily read by members of staff from all backgrounds and ages.
- 1.03. The layout of all parts of the system should be consistent with each other to make it easy to follow.
- 1.04. The application should be aesthetically pleasing to the eye and have a modern design with various colours and mechanics ensuring that it is not too colourful to be distracting or include too many mechanics that it is too complicated to use.
- 1.05. All data that is entered into the application must be validated before it is entered into the database to ensure that all data is entered and there are valid according to the system.

### 2. Main Menu

- 2.01. The Main Menu Form should act as the main hub and therefore must contain buttons to access the rest of the application to register and update a client, dog or staff record in the database as well as buttons to create and edit a booking of a grooming appointment and each of their payment details. A button to see payment reports should also be present.
- 2.02. A title must be present above every page and should change along with the button presses so the user can quickly identify what screen they are currently on.
- 2.03. The latest button that was clicked on the main menu must be highlighted as another way for the user to easily identify which screen is currently showing.
- 2.04. A toolbar must be present on top of each page containing buttons to the other pages inside the current screen. For instance, once the Register button is clicked, a toolbar containing Client, Dog and Staff should be displayed for the user to easily switch between registering a client to registering their dog etc.
- 2.05. The latest button pressed on the toolbar should be highlighted and colour-coded to the same colour as the current page displayed. (E.g. Client button should be highlighted when on Client page).
- 2.06. The user must be provided with a means to exit the application or minimise and so the main menu must host a minimise and close buttons.

### 3. Home

#### **4. Client**

- 4.01 The Client Screen must have two possible states: one for Registering/Adding client records and another for Updating/Editing existing client records in the database.
- 4.02 In the registering and updating screen, the user must be able to enter or change relevant details of the new client onto the data fields with each of them being labelled appropriately to ensure easy navigation.
- 4.03 The next available Client ID number must be automatically displayed when registering a new client to avoid errors and data inconsistency.
- 4.04 In the updating screen, there must be a way to search for existing clients and show their details on the data fields so that they can be updated. Buttons should be present to navigate through adjacent client records in order of their Client ID number.
- 4.05 In the updating screen, there must be a way for the user to delete an existing client after they have searched for them and a message should be shown notifying a successful operation.
- 4.06 In both the registering and updating screens, error providers must flash when invalid data is entered into the fields and the record must not be allowed to be registered or updated if any of its values is invalid. All fields will be required to be filled.
- 4.07 The Client ID number should always be in text format and have 4 characters. To avoid confusion and errors this text field should be set to read only to prohibit the user from changing it.
- 4.08 The Forename and Surname text fields must only contain letters and/or an apostrophe (e.g. O'Doherty) and should only allow a maximum of 30 characters. Errors with accompanying messages should be displayed if invalid.
- 4.09 The Sex text field should be a selection box so the user cannot enter anything other than 'Male', 'Female' or 'Other'. Errors should be displayed if invalid.
- 4.10 The user should not be able to enter clients that are under 18 years of age or an age that is unrealistic so this must be checked in the Date of Birth field. Errors should flash notifying the user if an invalid date is entered.
- 4.11 The Address text field should only allow letters and numbers and values entered should not exceed 50 characters otherwise errors will be displayed.
- 4.12 The City/Town text field should only allow letters and forward slashes ('/') and values with a max of 50 characters otherwise errors will be displayed.
- 4.13 The client's Postcode should be entered in the correct UK Postcode format ('LLNN NLL') to be passed through or errors will be revealed with a user-friendly message.
- 4.14 The E-mail text field should consist of a valid e-mail address (somenone@email.com), no more than 50 characters and containing no white spaces otherwise errors will be displayed notifying where the user went wrong.

- 4.15 The Phone Number text field should allow between 11 and 15 characters long consisting of only numbers and maybe a plus character ('+') at the start. Errors will be shown if invalid.
- 4.16 The Date Joined should be automatically set to the current date when registering a new client. It can be changed when updating however the user should not be able to set the date to a future date. Errors should be shown if this is the case.
- 4.17 In the registering screen, after a successful Client registration, the user should automatically be transferred to the Dog Registration page to add a dog owned by the new client.
- 4.18 In the updating screen, there should be a way to show and quickly go to the dogs owned by the current client displayed.

## **5. Dog**

- 5.01. The Dog Screen should have two possible states: one for Registering/Adding dog records and another for Updating/Editing details of existing dog records in the database.
- 5.02. In the registering and updating screen, the user must be able to enter or change relevant details of the new dog onto the data fields with each of them being labelled appropriately to ensure easy navigation.
- 5.03. The next available Dog ID number must be automatically displayed when registering a new dog to avoid errors and data inconsistency.
- 5.04. In the updating screen, there must be a way to search for existing dogs and show their details on the data fields so that they can be updated. Buttons should be present to navigate through adjacent dog records in order of their Dog ID number.
- 5.05. In the updating screen, there must be a way for the user to delete an existing dog after they have searched for them and a message should be shown notifying a successful deletion.
- 5.06. On the updating screen, there must be an image for each dog that is being displayed so the they can be easily be identified and searched for, plus it will give a professional look to the application.
- 5.07. In both the registering and updating screens, error providers must flash when invalid data is entered into the fields and the record must not be allowed to be registered or updated if any of its values is invalid. All fields will be prohibited to be left empty.
- 5.08. The Dog ID number should always be in text format and have 4 characters. To avoid confusion and errors this text field should be set to read only to prohibit the user from changing it.
- 5.09. The client/owner's name must only permit clients that are already registered into the system. Any foreign clients should elicit an error being displayed.
- 5.10. The name text field must only permit letters and a maximum of 30 characters or errors with messages will be displayed.
- 5.11. The breed text field should only permit letters and/or white spaces and a maximum of 30 characters otherwise error providers will be shown.

- 5.12. The sex text field should be a selection box so the user cannot enter anything other than 'Male' or 'Female'. Errors should be displayed if invalid.
- 5.13. The date of birth of the dog should be realistic and so the user should not be allowed to set it to a future date or a date 50 years before the current date. Errors should display if this happens.
- 5.14. The additional medical notes of the dog should not be left void and so if the user does not enter any value than it should be automatically set to 'None' to avoid any errors.
- 5.15. There must be a button to look at and update the dog's vaccination history listing all the vaccinations the dog has received and when.

## **6. Staff**

- 6.01. The Staff Screen should have two initial states: one for Registering/Adding staff records and another for Updating/Editing existing staff member records in the database.
- 6.02. In the registering and updating screen, the user must be able to enter or change relevant details of the new staff member onto the data fields with each of them being labelled appropriately to ensure easy navigation.
- 6.03. The next available Staff ID number must be automatically displayed when registering a new staff member to avoid errors and data inconsistency.
- 6.04. In the updating screen, there must be a way to search for existing members of staff and show their details on the data fields so that they can be updated. Buttons should be present to navigate through adjacent staff records in order of their Staff ID number.
- 6.05. In the updating screen, there must be a way for the user to delete an existing staff member after they have searched for them and a message should be shown notifying a successful operation.
- 6.06. In both the registering and updating screens, error providers must flash when invalid data is entered into the fields and the record must not be allowed to be registered or updated if any of its values is invalid. All fields will be required to be filled.
- 6.07. The Staff ID number should always be in text format and have 4 characters. To avoid confusion and errors this text field should be set to read only to prohibit the user from changing it.
- 6.08. The forename and surname text fields must only contain letters and/or an apostrophe (e.g. O'Doherty) and should only allow a maximum of 30 characters. Errors with accompanying messages should be displayed if invalid.
- 6.09. The sex text field should be a selection box so the user cannot enter anything other than 'Male', 'Female' or 'Other'. Errors should be displayed if not correct.
- 6.10. The user should not be able to enter clients that are under 16 years of age or an age that is unrealistic so this must be checked in the date of birth field. Errors should flare notifying the user if an invalid date is entered.
- 6.11. The address text field should only permit letters and numbers. Values entered should not exceed 50 characters otherwise errors will be displayed.
- 6.12. The city/town text field should only allow letters and forward slashes ("/") and values with a max of 50 characters otherwise errors will be shown.



- 6.13. The client's postcode should be entered in the correct UK Postcode format ('LLNN NLL') to be passed through or errors will be revealed with a helpful message.
- 6.14. The e-mail text field should consist of a valid e-mail address (somenone@email.com), no more than 50 characters and containing no white spaces otherwise errors will be displayed notifying where the user went wrong.
- 6.15. The phone number text field should allow between 11 and 15 characters long consisting of only numbers and maybe a plus character ('+') at the start. Errors will be shown if it does not follow these rules.
- 6.16. The user must tick one of the check boxes either 'Full-Time' or 'Part-Time' for the type of job contract the staff member currently is on or else errors will flash and prevent the user from carrying on.
- 6.17. The job position describes the specific job the staff members does and so the user will have to pick one of the following: 'Management', 'Administration' or 'Grooming' in order to continue without error.
- 6.18. The salary should only permit up to a maximum of 10 digits and should be stored to two decimal places. If the user sways away from these constraints, errors must be shown, and the user must be prevented from continuing further.
- 6.19. The date joined should be automatically set to the current date when registering a new staff employee. It can be changed when updating however the user should not be able to set the date to a future date. Errors should be shown if this is the case.

## **7. Booking**

- 7.01. The Booking Screen should have two initial forms: one for Creating/Adding new booking records and another for Updating/Editing existing booking records in the database.
- 7.02. In the create booking screen, the user must be able to enter relevant details of the new booking onto the data fields with each of them being labelled appropriately to ensure easy understanding.
- 7.03. The next available Booking Number must be automatically displayed when booking a new appointment to avoid errors and data inconsistency.
- 7.04. In the updating screen, there must be a way to search for existing bookings and show their details in a table/grid so that they can be selected with ease. Different bookings can be for the same dogs which can be confusing for the user.
- 7.05. In the updating screen, there must be a way for the user to delete an existing booking record after they have searched for them and a message should be shown notifying a successful operation.
- 7.06. In both the create booking and updating screens, error providers must flash when invalid data is entered into the fields and the record must not be allowed to be recorded or updated if any of its values is invalid. All fields will be required to be filled.
- 7.07. The Booking Number should always be in text format and have 4 characters. To avoid confusion and errors this text field should be set to read only to prohibit the user from changing it.

- 7.08. The user should be only permitted to enter names of dogs that already exist in the database and this should automatically fill the selection box for the client's name with the full names of clients that own a dog with this name. Both the dog's name and its owner's name values should be used to automatically search for their Dog ID number.
- 7.09. Under the Create Booking state, the user should only be allowed to set the date of the appointment in the future or the present date. Errors should be shown if this is disobeyed. Also, the user should not be able to book an appointment more than 6 months from the present.
- 7.10. The user should only be permitted to set the start time of the appointment within the working hours at JD Dog Care otherwise errors will display. Additionally, under the Create Booking state, if the date of the appointment is set to the present day, the start time should forbid times that are before the current time.
- 7.11. The service option for the booking should only allow the three options that JD Dog Care employ to be selected or errors must flash. The extra option for nails, teeth and ears should be check box where the user can select or deselect it depending on the client's choice. The duration and price for the booking should automatically update when the option is selected as well as when the extra option is checked or unchecked.
- 7.12. Under the Create Booking state, if this is the first booking for this client then the extra fee of 15 minutes and £5 should be added to the duration and price of the booking respectively.
- 7.13. The staff name should only permit staff members that work at JD Dog Care and their Staff ID number should be only automatically found and set to the current booking otherwise errors should display.
- 7.14. The date created should be automatically set to the current date when creating a new booking. It can be changed when updating however the user should not be able to set the date to a date after the date of the appointment. Errors should be shown if this is the case.

## **8. Payment**

- 8.01. The Payment Screen should have two possible states: one for searching for payment details about a particular booking and another for viewing and Creating/Updating a payment record to the database.

## **9. Vaccination**

## **10. Service Option**

## **11. Reports**

## Database Design

### Normalisation

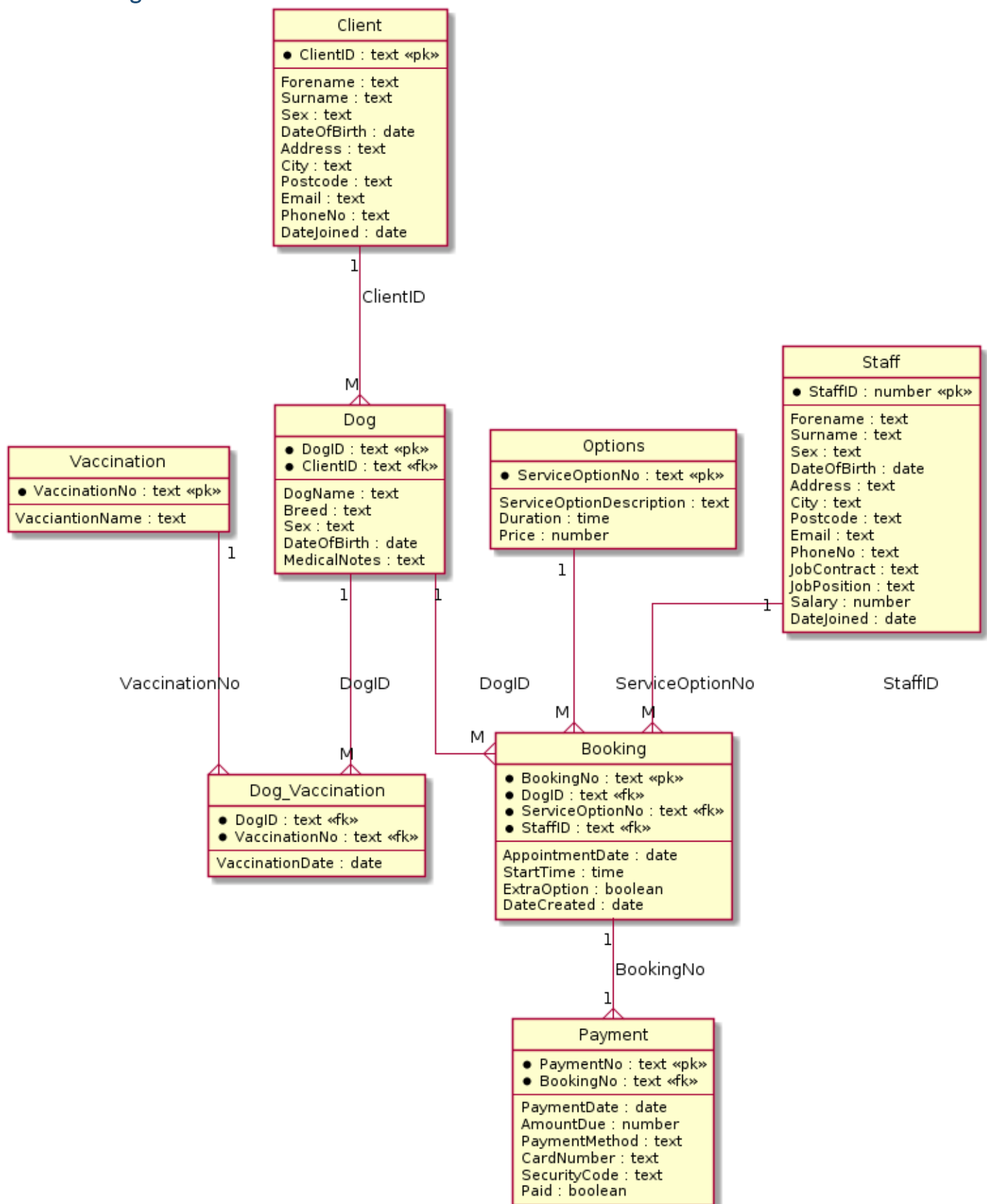
ONF:

**JDDOGCARE** (BookingNo, DogID, DName, DBreed, DSex, DDOB, DMedicalNotes,

VaccinationNo, VName, VDate, ClientID, CForename, CSurname, CSex, CDOB, CAddress, CCity, CPostcode, CEmail, CPhoneNo, CDateJoined, BDate, BStartTime, ServiceOption, SODescription, SODuration, SOPrice, BExtraOption, StaffID, SForename, SSurname, SSex, SDOB, SAddress, SCity, SPostcode, SEmail, SPhoneNo, SJobContract, SJobPostion, SSalary, SDateJoined)

## ER Diagram

## Dog Grooming ER Diagram



Stergios Aji 4211

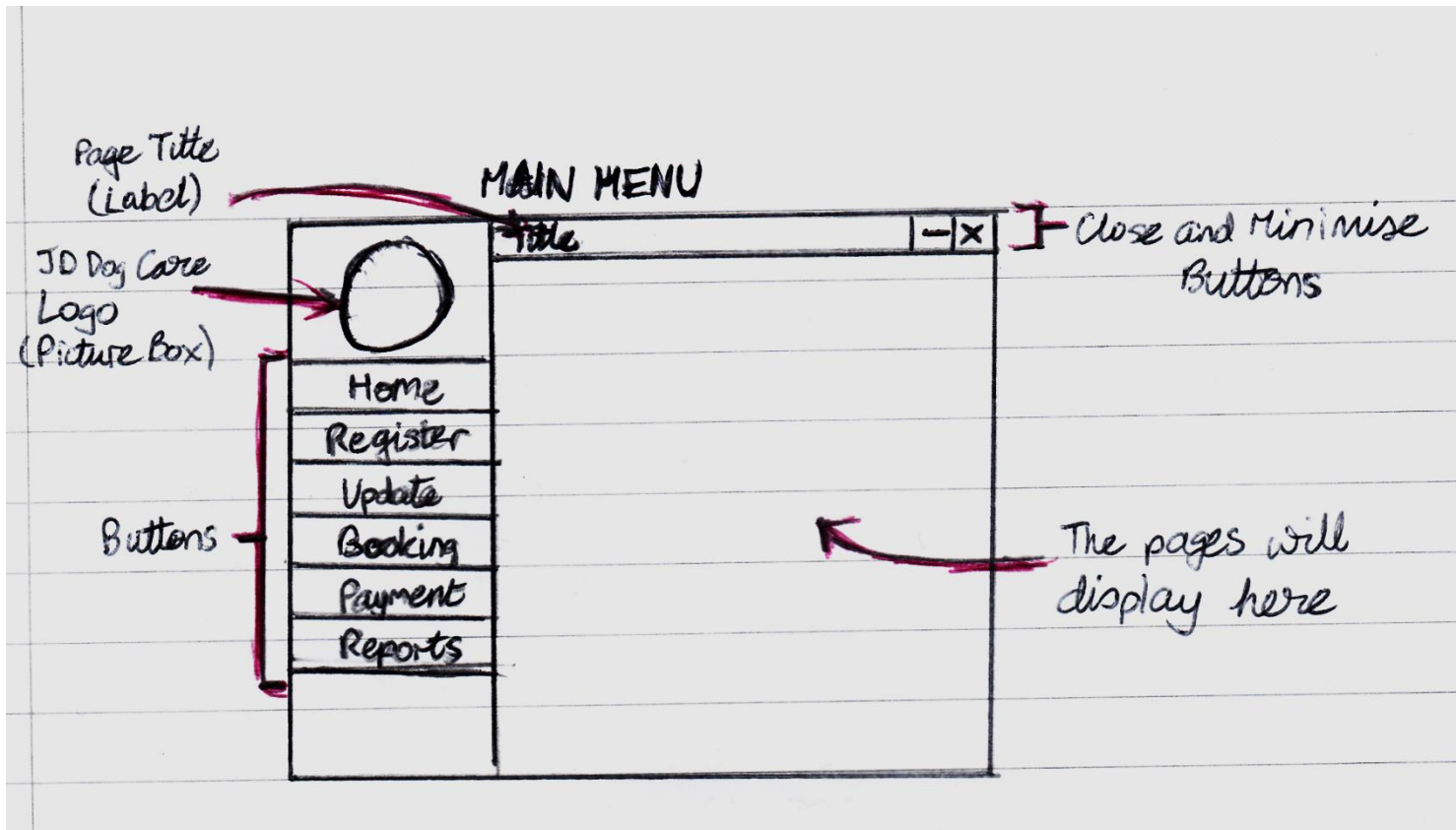
UML Diagram

Data Dictionary

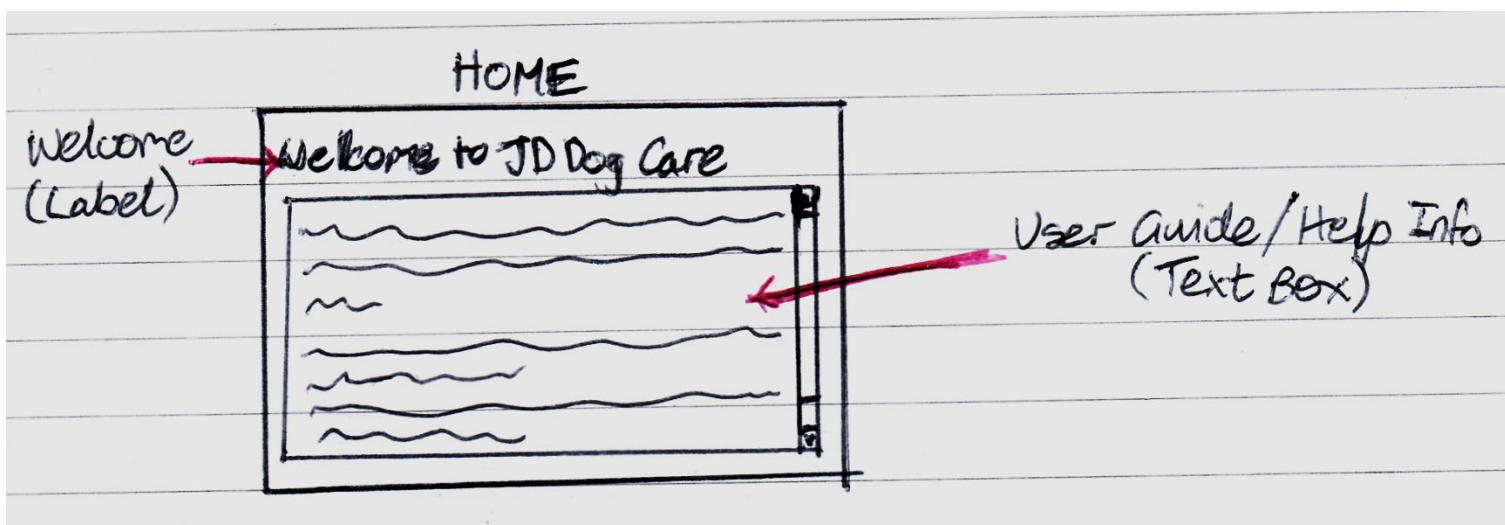
## UI Design

### Initial Sketches

#### Main Menu



#### Home





## Client

### - Register

**CLIENT REGISTRATION**

The form is titled "CLIENT REGISTRATION". It contains the following fields and controls:

- ClientID:** A text box with an error indicator (a circle with an 'X').
- City:** A text box.
- Forename:** A text box.
- Postcode:** A text box.
- Surname:** A text box.
- E-mail:** A text box.
- Sex:** A dropdown menu.
- DOB:** A date/time picker.
- Phone No.:** A text box.
- Address:** A text box.
- REGISTER:** A button.

Annotations with red arrows point to various components:

- Error Providers with captions:** Points to the ClientID field.
- Labels:** Points to the labels for Forename, Surname, Sex, DOB, and Address.
- Text Boxes:** Points to the City, Postcode, E-mail, and Phone No. fields.
- Combo Box:** Points to the Sex dropdown.
- Date Time Picker:** Points to the DOB field.
- Register Client Button:** Points to the REGISTER button.

### - Update

**CLIENT UPDATE**

The form is titled "CLIENT UPDATE". It includes a search bar at the top with navigation buttons (left and right arrows) and a search control. The main form contains the following fields and controls:

- ClientID:** A text box with an error indicator (a circle with an 'X').
- City:** A text box.
- Forename:** A text box.
- Postcode:** A text box.
- Surname:** A text box.
- E-mail:** A text box.
- Sex:** A dropdown menu.
- DOB:** A date/time picker.
- Phone No.:** A text box.
- Date Joined:** A date/time picker.
- Address:** A text box.
- UPDATE:** A button.
- DELETE:** A button.

Annotations with red arrows point to various components:

- Error Provider with captions:** Points to the ClientID field.
- Navigation buttons:** Points to the left and right arrows in the search bar.
- Search controls:** Points to the search bar.
- Text boxes:** Points to the City, Postcode, E-mail, Phone No., and Date Joined fields.
- Labels:** Points to the labels for Forename, Surname, Sex, DOB, and Address.
- Update Client Button:** Points to the UPDATE button.
- Delete Client button:** Points to the DELETE button.

## Dog

### - Register

**DOG REGISTRATION**

The form contains the following fields and controls:

- Labels:** DogID, Client Name, Name, Breed, Sex, DOB.
- Text Boxes:** Input fields for DogID, Client Name, Name, Breed, and DOB.
- Combo Boxes:** Dropdown menus for Client Name, Sex, and DOB.
- Error Provider:** A circle with an 'X' icon next to the DogID field.
- Date Time Picker:** A calendar icon next to the DOB field.
- Register Dog Button:** A button labeled 'REGISTER' at the bottom.

Annotations with red arrows point to these elements from the right side of the form.

### - Update

**DOG UPDATE**

The form contains the following fields and controls:

- Navigation Buttons:** Back and Forward arrow buttons at the top.
- Search Controls:** A search bar with a magnifying glass icon at the top.
- Error Provider:** A circle with an 'X' icon next to the DogID field.
- Check Boxes:** Checkboxes for 'Vaccinated' (Yes/No) and 'Client' (checked).
- Update Dog Button:** A button labeled 'UPDATE' at the bottom.
- Delete Dog Button:** A button labeled 'DELETE' at the bottom.

Other elements include a **Dog Profile (Picture Box)** showing a dog's head and input fields for DogID, ClientID, Client Name, Name, Breed, Sex, and DOB. Annotations with red arrows point to these elements from the left and right sides of the form.



## Staff

### - Register

**STAFF REGISTRATION**

StaffID: <input type="text"/>	<input checked="" type="radio"/>	Postcode: <input type="text"/>
Forename: <input type="text"/>		E-mail: <input type="text"/>
Surname: <input type="text"/>		Phone No: <input type="text"/>
Sex: <input type="text"/>		Contract: <input type="checkbox"/> Full time <input type="checkbox"/> Part time
DOB: <input type="text"/>		Position: <input type="text"/>
Address: <input type="text"/>		Salary: <input type="text"/>
City: <input type="text"/>		

**REGISTER**

**Annotations:**

- Labels: StaffID, Forename, Surname, Sex, DOB, Address, City
- Error Provider: Red circle with 'X' on StaffID
- Text Boxes: Postcode, E-mail, Phone No, Position, Salary
- Checkboxes: Full time, Part time
- Numeric Up & Down: Salary
- Register Staff Button: REGISTER

### - Update

**STAFF UPDATE**

<input type="text"/>	<input type="text"/>	<input type="text"/>
StaffID: <input type="text"/>	<input checked="" type="radio"/>	Postcode: <input type="text"/>
Forename: <input type="text"/>		E-mail: <input type="text"/>
Surname: <input type="text"/>		Phone No: <input type="text"/>
Sex: <input type="text"/>		Contract: <input type="checkbox"/> Full time <input type="checkbox"/> Part time
DOB: <input type="text"/>		Position: <input type="text"/>
Address: <input type="text"/>		Salary: <input type="text"/>
City: <input type="text"/>		Date Joined: <input type="text"/>

**UPDATE**      **DELETE**

**Annotations:**


- Error Providers: Red circle with 'X' on StaffID
- Navigation Buttons: Back, Forward
- Search Controls: Search bar
- Date Time Picker: Date Joined
- Delete Staff Button: DELETE
- Update Staff Button: UPDATE


## Booking


### - Create


**CREATE BOOKING**


Error Provider


Booking No:  

Day Name:  

Client Name:  

Date:  


Start Time:  

Service Option:  

☐ Extra Option

Duration:

Price:

Staff Name:  

Will update when a new option is selected

Create Booking Button

Date Time Picker in time format

Combo Box

Check box

### - Update


**UPDATE BOOKING**


Search Controls


DOUBLE-CLICK


Data Grid View showing all bookings


Error Provider


Booking No:  

Dog Name:  

Client Name:  

Date:  


Start Time:  


Service Option:  

☐ Extra Option

Duration:

Price:

Staff Name:  

Date Created:  

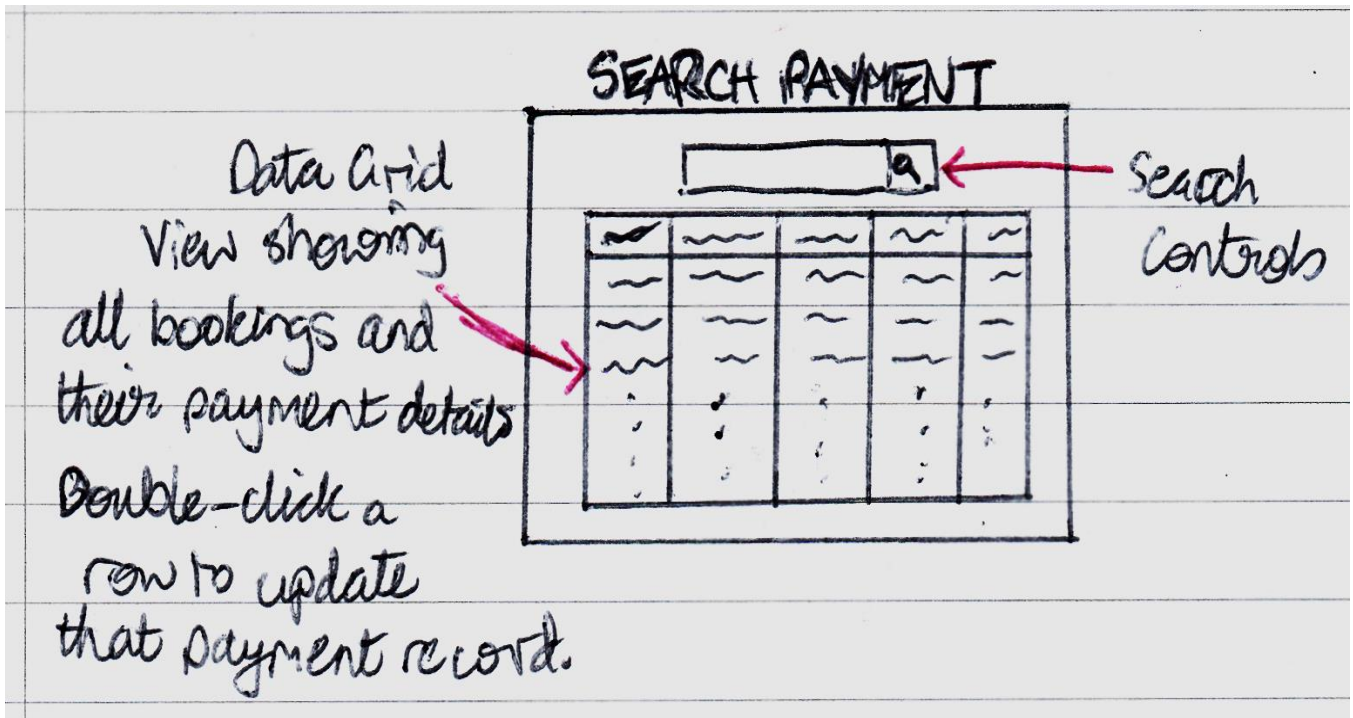
Update Booking Button

Delete Booking Button

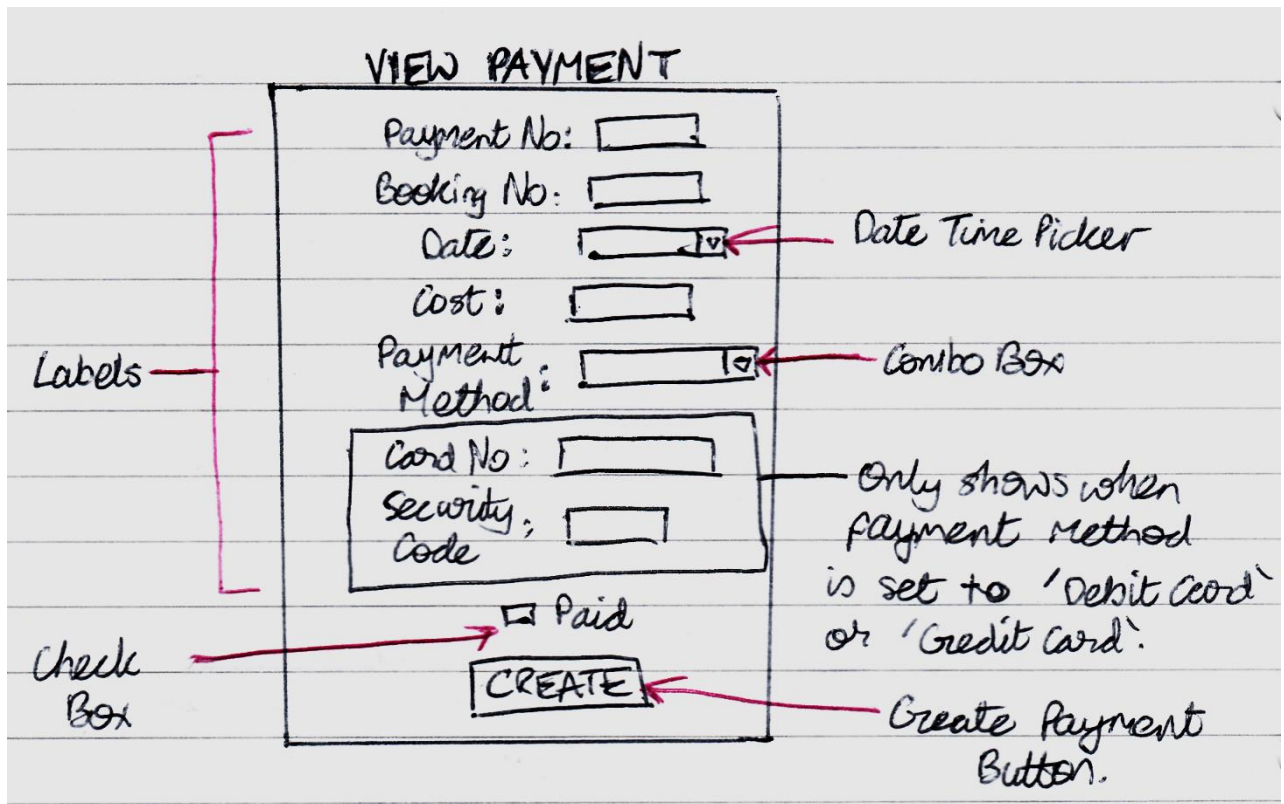


## Payment

### - Search

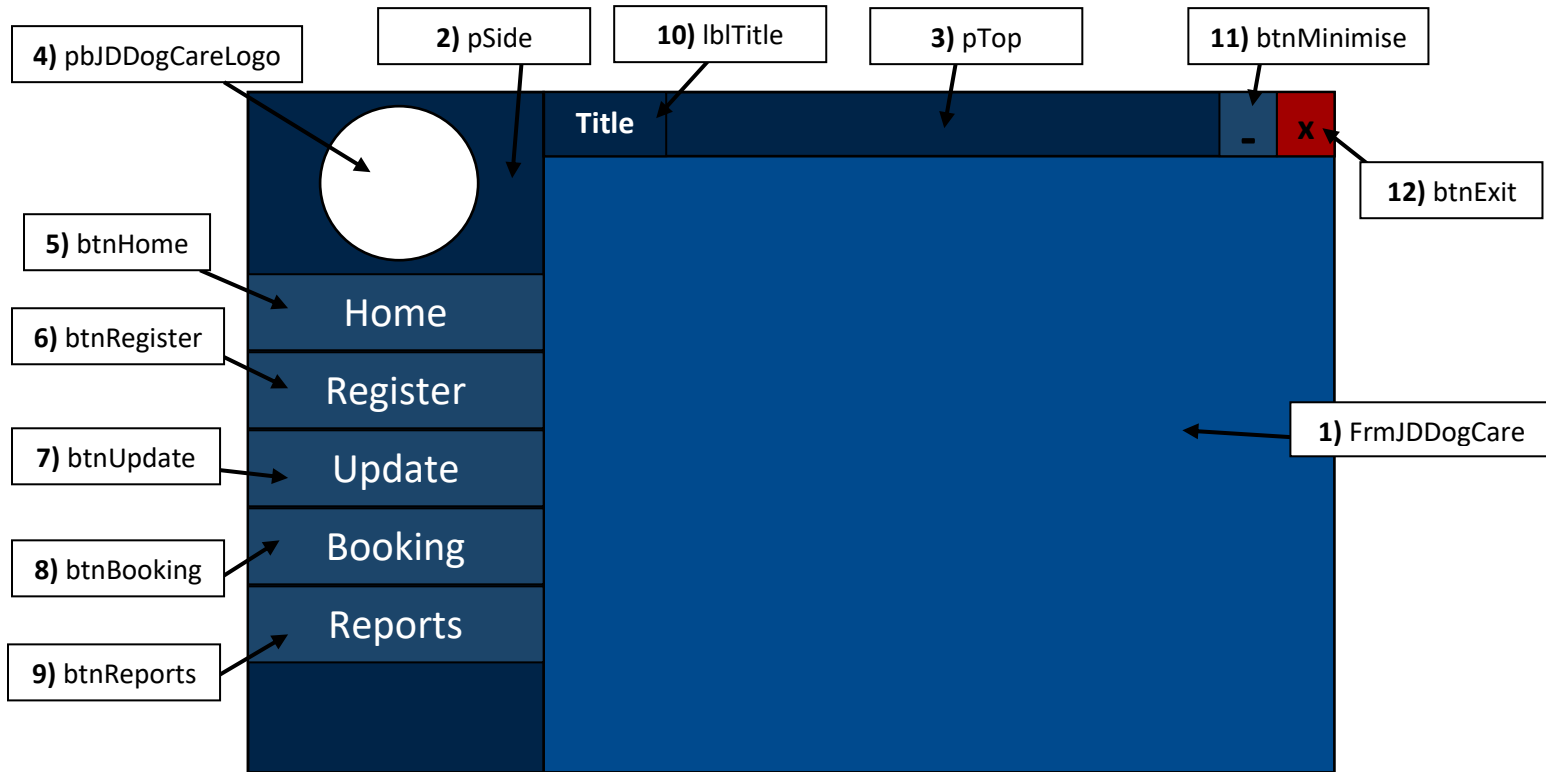


### - View



## Wireframes

### Main Menu



All text present in the application will have font 'Century Gothic' to keep consistency.

- 1) **FrmJDDogCare:** This will be the main menu screen and the first screen the user will be greeted with. All parts of the application will be accessed through and displayed here.
- 2) **pSide:** This side panel will hold the JD Dog Care Logo and all the buttons to access the other forms.
- 3) **pTop:** This top panel will hold the title and the minimise and close buttons.
- 4) **pbJDDogCareLogo:** This will display the company logo.
- 5) **btnHome:** This button will display the home page onto the form.
- 6) **btnRegister:** This button will display the registering screen to register a record.
- 7) **btnUpdate:** This button will display the updating screen to update a record.
- 8) **btnBooking:** This button will display the booking screen to create or update a booking record.
- 9) **btnReports:** This button will display the reports.
- 10) **lblTitle:** This label will display the title of the current page (e.g. Booking).
- 11) **btnMinimise:** This button will minimise the application to the taskbar.
- 12) **btnExit:** This button will completely close the application.

Control	Properties	Value	Events
Form	Name FormBorderStyle Size StartPosition	FrmJDDogCare None 1000 x 620 (px) CentreScreen	N/A
Panel	Name Size	pSide 240 x 440 (px)	N/A
Panel	Name Size	pTop 760 x 40 (px)	N/A
Picture Box	Name Size	pbJDDogCareLogo 240 x 220 (px)	N/A
Button	Name Text Cursor FlatStyle	btnHome Home Hand Flat	BtnHome_Click
Button	Name Text Cursor FlatStyle	btnRegister Register Hand Flat	BtnRegister_Click
Button	Name Text Cursor FlatStyle	btnUpdate Update Hand Flat	BtnUpdate_Click
Button	Name Text Cursor FlatStyle	btnBooking Booking Hand Flat	BtnBooking_Click
Button	Name Text Cursor FlatStyle	btnReports Reports Hand Flat	BtnReports_Click
Label	Name Text	lblTitle Home	N/A
Button	Name Text FlatStyle	btnMinimise - Flat	BtnMinimise_Click
Button	Name Text FlatStyle	btnExit × Flat	BtnExit_Click



Home



## Client

### - Register

1) UcClient

2) lblClientID  
3) txtClientID

4) lblForename  
5) txtForename

6) lblSurname  
7) txtSurname

8) lblSex  
9) cbSex

10) lblDOB  
11) dtpDOB

12) lblAddress  
13) txtAddress

ClientID:

Forename:

Surname:

Sex:

DOB:

Address:

City/Town:

Postcode:

E-mail:

Phone No:

Date Joined:

14) lblCity  
15) txtCity

16) lblPostcode  
17) txtPostcode

18) lblEmail  
19) txtEmail

20) lblPhoneNo  
21) txtPhoneNo

22) lblDateJoined  
23) dtpDateJoined

Register

24) btnClient

25) ep

**1) UcClient:** This User Control will be displayed when user clicks the Register button on the main menu and then the Client button.

**2), 4), 6), 8), 10), 12), 14), 16), 18), 20), 22) Labels:** Indicates what the text field adjacent is for.

**3), 5), 7), 13), 15), 17), 19), 21) Text Boxes:** User will have to enter data about the client into these text fields.

**9) cbSex:** Combo box where user will have to select the client's sex out of the options: Male, Female and Other.

**11), 23) Date Time Pickers:** The user will have to select the date of birth of the client on an interactive calendar. The date joined will already be set to the current date so will be disabled so the user cannot change it.

**24) btnClient:** Button that will validate and then register a client record holding all the data entered to the database.

**25) ep:** Error provider that will display on input fields that are invalid.

Control	Properties	Value	Events
User Control	Name Size	UcClient 760 x 550 (px)	N/A
Label	Name Text	lblClientID ClientID:	N/A
Text Box	Name ReadOnly	txtClientID True	TxtClientID_TextChanged
Label	Name Text	lblForename Forename:	N/A
Text Box	Name	txtForename	TxtForename_TextChanged
Label	Name Text	lblSurname Surname:	N/A
Text Box	Name	txtSurname	TxtSurname_TextChanged
Label	Name Text	lblSex Sex:	N/A
Combo Box	Name Items	cbSex [Male, Female, Other]	CbSex_TextChanged
Label	Name Text	lblDOB Date Of Birth:	N/A
Date Time Picker	Name Format	dtpDOB Short	DtpDOB_ValueChanged
Label	Name Text	lblAddress Address:	N/A
Text Box	Name	txtAddress	TxtAddress_TextChanged
Label	Name Text	lblCity City/Town:	N/A
Text Box	Name	txtCity	TxtCity_TextChanged
Label	Name Text	lblPostcode Postcode:	N/A
Text Box	Name	txtPostcode	TxtPostcode_TextChanged
Label	Name Text	lblEmail E-mail:	N/A
Text Box	Name	txtEmail	TxtEmail_TextChanged
Label	Name Text	lblPhoneNo Phone No:	N/A
Text Box	Name	txtPhoneNo	TxtPhoneNo_TextChanged
Label	Name Text	lblDateJoined Date Joined:	N/A
Date Time Picker	Name Format	dtpDateJoined Short	DtpDateJoined_ValueChanged
Button	Name Text Cursor	btnClient Register Hand	BtnClient_Click

	FlatStyle	Flat	
Error Provider	Name	ep	N/A

- Update

**1) UcClient:** This User Control will be displayed when user clicks the Update button on the main menu and then the Client button.

**2) – 25) Refer to page ~16~**

**26), 27) Search Text Box and Button:** Controls that can be used to search for a client by inputting their surname and forename

**28), 29) Next and Previous Buttons:** Buttons that can be used to navigate through all the clients in order of their ClientID.

**30) btnViewClientDogs:** Button that will display a table [30] with all the dogs that the current client owns.

**31) btnDelete:** Button that will delete the current client displayed and their dogs from database

**32) dgvDogsOwned:** Data Grid View that will be displayed when btnViewDogsOwned [30] is clicked. It will show a table with all the dogs owned by the client. The dog records can be double clicked to quickly transfer to its update screen.

Control	Properties	Value	Events
User Control	Name Size	UcClient 760 x 550 (px)	N/A
Label	Name Text	lblClientID ClientID:	N/A
Text Box	Name ReadOnly	txtClientID True	TxtClientID_TextChanged
Label	Name Text	lblForename Forename:	N/A
Text Box	Name	txtForename	TxtForename_TextChanged
Label	Name Text	lblSurname Surname:	N/A
Text Box	Name	txtSurname	TxtSurname_TextChanged
Label	Name Text	lblSex Sex:	N/A
Combo Box	Name Items	cbSex [Male, Female, Other]	CbSex_TextChanged
Label	Name Text	lblDOB Date Of Birth:	N/A
Date Time Picker	Name Format	dtpDOB Short	DtpDOB_ValueChanged
Label	Name Text	lblAddress Address:	N/A
Text Box	Name	txtAddress	TxtAddress_TextChanged
Label	Name Text	lblCity City/Town:	N/A
Text Box	Name	txtCity	TxtCity_TextChanged
Label	Name Text	lblPostcode Postcode:	N/A
Text Box	Name	txtPostcode	TxtPostcode_TextChanged
Label	Name Text	lblEmail E-mail:	N/A
Text Box	Name	txtEmail	TxtEmail_TextChanged
Label	Name Text	lblPhoneNo Phone No:	N/A
Text Box	Name	txtPhoneNo	TxtPhoneNo_TextChanged
Label	Name Text	lblDateJoined Date Joined:	N/A
Date Time Picker	Name Format	dtpDateJoined Short	DtpDateJoined_ValueChanged
Button	Name Text	btnClient Update	BtnClient_Click

	Cursor FlatStyle	Hand Flat	
Error Provider	Name	ep	N/A
Text Box	Name AutoCompleteMode	txtSearch SuggestAppend	N/A
Button	Name Icon	btnSearch [Search Icon]	BtnSearch_Click
Button	Name Text	btnPrevious ◀	BtnPrevoius_Click
Button	Name Text	btnNext ▶	BtnNext_Click
Button	Name Text	btnViewClientDogs View Dogs Owned	BtnViewClientDogs_Click
Button	Name Text	btnDelete Delete Client	BtnDelete_Click
Data Grid View	Name Location Size Visible	dgvDogsOwned 20 x 85 (px) 720 x 360 (px) False	DgvDogsOwned_CellContent DoubleClick



## Dog

### - Register

The diagram shows a 'Dog Register' form with the following elements and their corresponding labels:

- 1) UcDog**: Points to the entire form container.
- 2) lblDogID** and **3) txtDogID**: Point to the 'DogID:' label and its text input field.
- 4) lblClientName** and **5) cbClientName**: Point to the 'Client Name:' label and its dropdown menu.
- 6) lblName** and **7) txtName**: Point to the 'Name:' label and its text input field.
- 8) lblBreed** and **9) txtBreed**: Point to the 'Breed:' label and its text input field.
- 10) lblSex** and **11) cbSex**: Point to the 'Sex:' label and its dropdown menu.
- 12) lblDOB** and **13) dtpDOB**: Point to the 'DOB:' label and its date picker.
- 14) btnDog**: Points to the yellow 'Register' button.
- 15) ep**: Points to a red 'X' error provider icon.

- 1) UcDog:** This User Control will be display when the user clicks the Register button on the main menu screen and then the Dog button.
- 2), 4), 6), 8), 10), 12) Labels:** Indicate the what the text field adjacent is for.
- 3), 7), 9) Text Boxes:** User must enter data about the client into these text fields.
- 5) cbClientName:** Combo box will hold all the client's names currently in the database and the user will simply have to choose the appropriate client that owns this new dog.
- 11) cbSex:** Combo box where user will have to select the dog's sex out of the options: Male and Female.
- 13) dtpDOB:** The user will have to select the date of birth of the dog on an interactive calendar.
- 17) btnDog:** This button will validate and then register a dog record holding all the data entered to the text fields.
- 18) ep:** Error provider that will display on input fields that are invalid.

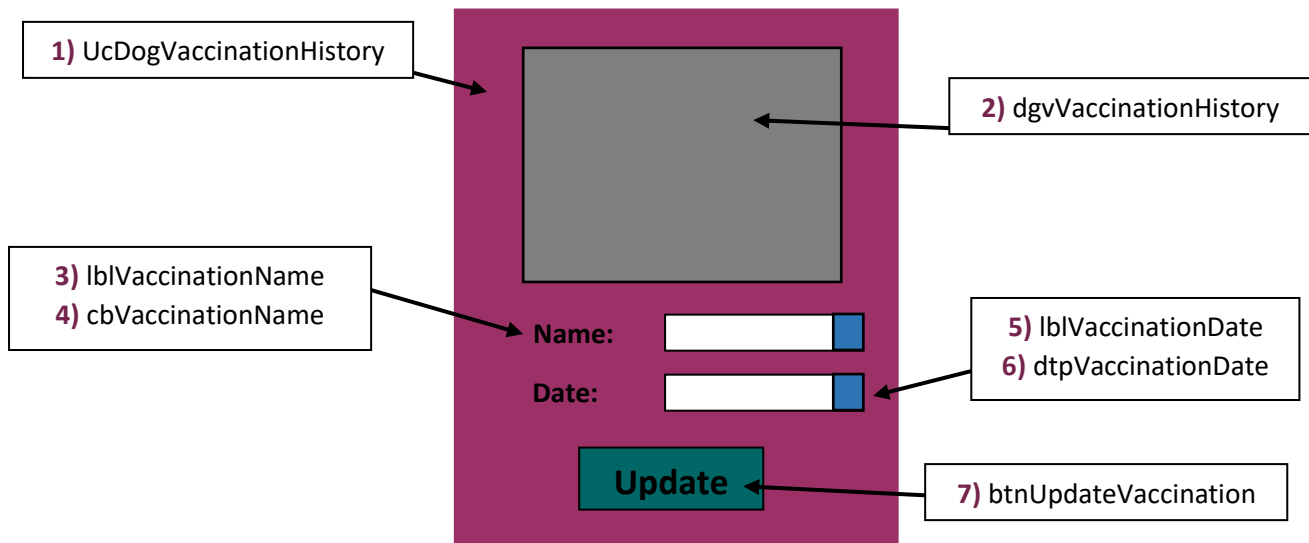
Control	Properties	Value	Events
User Control	Name Size	UcDog 760 x 550 (px)	N/A
Label	Name Text	lblDogID DogID:	N/A
Text Box	Name ReadOnly	txtDogID True	TxtDogID_TextChanged
Label	Name Text	lblClientName Client Name:	N/A
Combo Box	Name	cbClientName	CbClientName_TextChanged
Label	Name Text	lblName Name:	N/A
Text Box	Name	txtName	TxtName_TextChanged
Label	Name Text	lblBreed Breed:	N/A
Text Box	Name	txtBreed	TxtBreed_TextChanged
Label	Name Text	lblSex Sex:	N/A
Combo Box	Name Items	cbSex [Male, Female]	CbSex_TextChanged
Label	Name Text	lblDOB Date Of Birth:	N/A
Date Time Picker	Name Format	dtpDOB Short	DtpDOB_ValueChanged
Button	Name Text Cursor FlatStyle	btnDog Register Hand Flat	BtnDog_Click
Error Provider	Name	ep	N/A



Control	Properties	Value	Events
User Control	Name Size	UcDog 760 x 550 (px)	N/A
Label	Name Text	lblDogID DogID:	N/A
Text Box	Name ReadOnly	txtDogID True	TxtDogID_TextChanged
Label	Name Text	lblClientName Client Name:	N/A
Combo Box	Name	cbClientName	CbClientName_TextChanged
Label	Name Text	lblName Name:	N/A
Text Box	Name	txtName	TxtName_TextChanged
Label	Name Text	lblBreed Breed:	N/A
Text Box	Name	txtBreed	TxtBreed_TextChanged
Label	Name Text	lblSex Sex:	N/A
Combo Box	Name Items	cbSex [Male, Female]	CbSex_TextChanged
Label	Name Text	lblDOB Date Of Birth:	N/A
Date Time Picker	Name Format	dtpDOB Short	DtpDOB_ValueChanged
Button	Name Text Cursor FlatStyle	btnDog Register Hand Flat	BtnDog_Click
Error Provider	Name	ep	N/A
Label	Name Text	lblClientID ClientID:	N/A
Text Box	Name ReadOnly	txtClientID True	TxtClientID_TextChanged
Button	Name Text	btnViewVaccinationHistory View Vaccination History	BtnViewVaccinationHistory_Click
Label	Name Text	lblMedicalNotes Medical Notes:	N/A
Rich Text Box	Name Size	rtxtMedicalNotes 265 x 220 (px)	RtxtMedicalNotes_TextChanged
Picture Box	Name Size	pbDogProfile 265 x 280 (px)	N/A
Text Box	Name	txtSearch	N/A

	AutoComplete Mode	SuggestAppend	
Button	Name Icon	btnSearch [Search Icon]	BtnSearch_Click
Button	Name Text	btnPrevious ◀	BtnPrevoius_Click
Button	Name Text	btnNext ▶	BtnNext_Click
Button	Name Text	btnViewMedicalNotes View Dogs Owned	BtnViewMedicalNotes _Click
Button	Name Text	btnDelete Delete Dog	BtnDelete_Click

## Dog Vaccination History



- 1) **UcDogVaccinationHistory:** This User Control will be display when the user clicks the View Vaccination History button on the Update Dog page.
- 2) **dgvVaccinationHistory:** This Data Grid View will show all the vaccinations that the current dog has received as well as the date received. It can be double clicked to quickly fill the input fields for a vaccination the user wants to update.
- 3), 5) **Labels:** Indicate the what the input field adjacent is for.
- 4) **cbVaccinationName:** This Combo Box will hold all the vaccination names that is mandatory for a dog to receive.
- 6) **dtpVaccinationDate:** This Date Time Picker can be set to a date when the dog had received the named vaccination in cbVaccinationName [4)]. This will set to 'NULL' if the dog has not received that particular vaccination.
- 7) **btnUpdateVaccination:** This Button, when clicked will check if the user input is valid and if it is then it will update the dog's vaccination history.

Control	Properties	Value	Events
User Control	Name Size	UcDogVaccinationHistory 463 x 415 (px)	N/A
Data Grid View	Name Location Size	dgvVaccinationHistory 25 x 25 (px) 413 x 200 (px)	DgvVaccinationHistory_ CellContentDoubleClick
Label	Name Text	lblVaccinationName Vaccination Name:	N/A
Combo Box	Name	cbVaccinationName	CbVaccinationName_ TextChanged
Label	Name Text	lblVaccinationDate Vaccination Date:	N/A
Date Time Picker	Name	dtpVaccinationDate	DtpVaccinationDate_ ValueChanged
Button	Name Text	btnUpdateVaccination Update	BtnUpdateVaccination_ Click

## Staff

### - Register

**1) UcStaff:** This User Control will be displayed when user clicks the Register button on the main menu and then the Staff button.

**2), 4), 6), 8), 10), 12), 14), 16), 18), 20), 22), 25), 27), 29) Labels:** Indicates what the text field adjacent is for.

**3), 5), 7), 13), 17), 19), 21), 26) Text Boxes & cbJobPosition:** User will have to enter data about the employee into these text fields.

**9) cbSex:** Combo box where user will have to select the employee's sex out of the options: Male, Female and Other.

**11), 30) Date Time Pickers:** The user will have to select the date of birth of the staff member on an interactive calendar. The date joined will already be set to the current date so will be disabled so the user cannot change it.

**23), 24) Check Boxes:** The user will have to check or uncheck 'Full-Time contract' or 'Part-Time contract' to describe the type of contract the employee is currently on.

**28) nupSalary:** Numeric Up/Down for the user to change or type into the current salary of a staff member.

**31) btnStaff:** Button that will validate and then register a staff record holding all the data entered to the database.

**32) ep:** Error provider that will display on input fields that are invalid.



Control	Properties	Value	Events
User Control	Name Size	UcStaff 760 x 550 (px)	N/A
Label	Name Text	lblStaffID StaffID:	N/A
Text Box	Name ReadOnly	txtStaffID True	TxtStaffID_TextChanged
Label	Name Text	lblForename Forename:	N/A
Text Box	Name	txtForename	TxtForename_TextChanged
Label	Name Text	lblSurname Surname:	N/A
Text Box	Name	txtSurname	TxtSurname_TextChanged
Label	Name Text	lblSex Sex:	N/A
Combo Box	Name Items	cbSex [Male, Female, Other]	CbSex_TextChanged
Label	Name Text	lblDOB Date Of Birth:	N/A
Date Time Picker	Name Format	dtpDOB Short	DtpDOB_ValueChanged
Label	Name Text	lblAddress Address:	N/A
Text Box	Name	txtAddress	TxtAddress_TextChanged
Label	Name Text	lblCity City/Town:	N/A
Text Box	Name	txtCity	TxtCity_TextChanged
Label	Name Text	lblPostcode Postcode:	N/A
Text Box	Name	txtPostcode	TxtPostcode_TextChanged
Label	Name Text	lblEmail E-mail:	N/A
Text Box	Name	txtEmail	TxtEmail_TextChanged
Label	Name Text	lblPhoneNo Phone No:	N/A
Text Box	Name	txtPhoneNo	TxtPhoneNo_TextChanged
Label	Name Text	lblJobContract Contract:	N/A
Check Box	Name Text FlatStyle	chbFullTime Full-Time Flat	ChbFullTime_Click
Check Box	Name Text	chbPartTime Part-Time	ChbPartTime_Click

	FlatStyle	Flat	
Label	Name Text	lblJobPosition Position:	N/A
Text Box	Name	txtPosition	TxtPosition_TextChanged
Label	Name Text	lblSalary Salary:	N/A
Numeric Up Down	Name DecimalPlaces	nupSalary 2	NupSalary_ValueChanged
Label	Name Text	lblDateJoined Date Joined:	N/A
Date Time Picker	Name Format	dtpDateJoined Short	DtpDateJoined_ValueChanged
Button	Name Text Cursor FlatStyle	btnStaff Register Hand Flat	BtnStaff_Click
Error Provider	Name	ep	N/A

- Update

1) UcStaff

2) lblStaffID  
3) txtStaffID

4) lblForename  
5) txtForename

6) lblSurname  
7) txtSurname

8) lblSex  
9) cbSex

10) lblDOB  
11) dtpDOB

12) lblAddress  
13) txtAddress

14) lblCity  
15) txtCity

16) lblPostcode  
17) txtPostcode

18) lblEmail  
19) txtEmail

20) lblPhoneNo  
21) txtPhoneNo

22) lblJobContract  
23) chbFullTime  
24) chbPartTime

25) lblJobPosition  
26) cbJobPosition

27) lblSalary  
28) nupSalary

29) lblDateJoined  
30) dtpDateJoined

31) btnStaff

32) ep

33) txtSearch

34) btnSearch

35) btnPrevious

36) btnNext

37) btnDelete

Update

Delete Staff

**1) UcStaff:** This User Control will be displayed when user clicks the Update button on the main menu and then the Staff button.

**2) – 32) Refer to page ~26~**

**26), 27) Search Text Box and Button:** Controls that can be used to search for an employee by inputting their surname and forename

**28), 29) Next and Previous Buttons:** Buttons that can be used to navigate through all the staff members in order of their ClientID.

**37) btnDelete:** Button that will delete the current staff displayed from the database.

Control	Properties	Value	Events
User Control	Name Size	UcStaff 760 x 550 (px)	N/A
Label	Name Text	lblStaffID StaffID:	N/A
Text Box	Name ReadOnly	txtStaffID True	TxtStaffID_TextChanged
Label	Name Text	lblForename Forename:	N/A
Text Box	Name	txtForename	TxtForename_TextChanged
Label	Name Text	lblSurname Surname:	N/A
Text Box	Name	txtSurname	TxtSurname_TextChanged
Label	Name Text	lblSex Sex:	N/A
Combo Box	Name Items	cbSex [Male, Female, Other]	CbSex_TextChanged
Label	Name Text	lblDOB Date Of Birth:	N/A
Date Time Picker	Name Format	dtpDOB Short	DtpDOB_ValueChanged
Label	Name Text	lblAddress Address:	N/A
Text Box	Name	txtAddress	TxtAddress_TextChanged
Label	Name Text	lblCity City/Town:	N/A
Text Box	Name	txtCity	TxtCity_TextChanged
Label	Name Text	lblPostcode Postcode:	N/A
Text Box	Name	txtPostcode	TxtPostcode_TextChanged
Label	Name Text	lblEmail E-mail:	N/A
Text Box	Name	txtEmail	TxtEmail_TextChanged
Label	Name Text	lblPhoneNo Phone No:	N/A
Text Box	Name	txtPhoneNo	TxtPhoneNo_TextChanged
Label	Name Text	lblJobContract Contract:	N/A
Check Box	Name Text FlatStyle	chbFullTime Full-Time Flat	ChbFullTime_Click
Check Box	Name	chbPartTime	ChbPartTime_Click

	Text FlatStyle	Part-Time Flat	
Label	Name Text	lblJobPosition Position:	N/A
Text Box	Name	txtPosition	TxtPosition_TextChanged
Label	Name Text	lblSalary Salary:	N/A
Numeric Up Down	Name DecimalPlaces	nupSalary 2	NupSalary_ValueChanged
Label	Name Text	lblDateJoined Date Joined:	N/A
Date Time Picker	Name Format	dtpDateJoined Short	DtpDateJoined_ValueChanged
Button	Name Text Cursor FlatStyle	btnStaff Register Hand Flat	BtnStaff_Click
Error Provider	Name	ep	N/A
Text Box	Name AutoCompleteMode	txtSearch SuggestAppend	N/A
Button	Name Icon	btnSearch [Search Icon]	BtnSearch_Click
Button	Name Text	btnPrevious ◀	BtnPrevoius_Click
Button	Name Text	btnNext ▶	BtnNext_Click
Button	Name Text	btnDelete Delete Staff	BtnDelete_Click

## Booking

### - Create

The diagram illustrates the 'Create Booking' user control. It includes the following elements and their corresponding labels:

- 1) UcBooking**: The main user control container.
- 2) lblBookingNo** and **3) txtBookingNo**: Labels and text box for the booking number.
- 4) lblDogName** and **5) cbDogName**: Label and combo box for the dog's name.
- 6) lblClientName** and **7) cbClientName**: Label and combo box for the client's name.
- 8) lblDate** and **9) dtpDate**: Label and date/time picker for the appointment date.
- 10) lblStartTime** and **11) dtpStartTime**: Label and date/time picker for the appointment start time.
- 12) lblServiceOption** and **13) cbServiceOption**: Label and combo box for the service option.
- 14) chbExtraOption**: Check box for 'Add Extra Option'.
- 15) lblDuration** and **16) txtDuration**: Label and text box for the service duration.
- 17) lblPrice** and **18) txtPrice**: Label and text box for the service price.
- 19) lblStaffName** and **20) cbStaffName**: Label and combo box for the staff member.
- 21) lblDateCreated** and **22) dtpDateCreated**: Label and date/time picker for the date the booking was created.
- 23) btnBooking**: The 'Create Booking' button.
- 24) ep**: An error provider (marked with a red X) for invalid input.

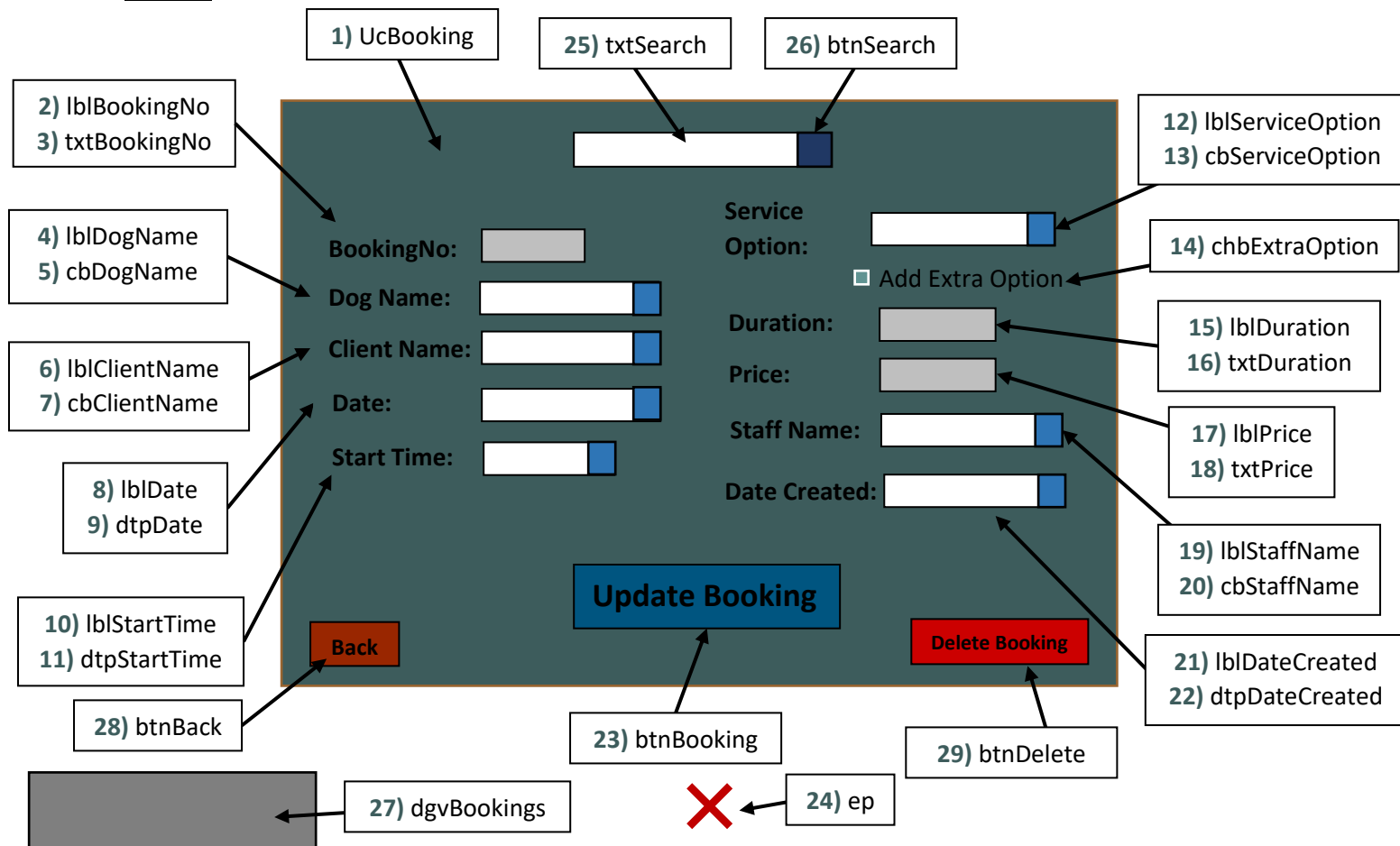
- 1) UcBooking:** This User Control will be displayed when user clicks the Booking button on the main menu and then the Create Booking button.
- 2), 4), 6), 8), 10), 12), 15), 17), 19), 22) Labels:** Indicates what the input field adjacent is for.
- 3) txtBookingNo:** Read only text field that is already set to the next available BookingNo in the database table when the screen is shown.
- 5) cbDogName:** Combo box where the user will have to select the dog's name.
- 7) cbClientName:** Combo box where the user will have to select the client's name.
- 11) dtpDate:** This will hold the date of the appointment.
- 12) dtpStartTime:** This will hold the start time of the appointment
- 13) cbServiceOption:** Combo box where the user will have to select the service option for the appointment.
- 14) chbExtraOption:** User can check it to include extra fee for incorporating nails, teeth and ears into the booking.
- 16) txtDuration:** Displays the duration of the selected service option.
- 18) txtPrice:** Displays the price of the selected service option.
- 20) cbStaffName:** Combo box holding the staff member who will take care of this booking.
- 22) dtpDateCreated:** Date Time Picker should automatically be set to the current date.
- 23) btnBooking:** Button that will validate and then create and add a booking holding all the data entered to the database.
- 24) ep:** Error provider that will display on input fields that are invalid.

Control	Properties	Value	Events
User Control	Name Size	UcBooking 760 x 550 (px)	N/A
Label	Name Text	lblBookingNo BookingNo:	N/A
Text Box	Name ReadOnly	txtBookingNo True	TxtBookingNo_TextChanged
Label	Name Text	lblDogName Dog Name:	N/A
Combo Box	Name	cbDogName	CbDogName_TextChanged
Label	Name Text	lblClientName Client Name:	N/A
Combo Box	Name	cbClientName	CbClientName_TextChanged
Label	Name Text	lblDate Date:	N/A
Date Time Picker	Name Format	dtpDate Short	DtpDate_ValueChanged
Label	Name Text	lblStartTime Start Time:	N/A
Date Time Picker	Name CustomFormat	dtpStartTime 'hh:mm tt'	DtpStartTime_ValueChanged
Label	Name Text	lblServiceOption ServiceOption:	N/A
Combo Box	Name Items	cbServiceOption [Option 1, Option 2, Option 3]	CbServiceOption_TextChanged
Check Box	Name Text	chbExtraOption Add Extra Option	ChbExtraOption_CheckStateChanged
Label	Name Text	lblDuration Duration:	N/A
Text Box	Name ReadOnly	txtDuration True	TxtDuration_TextChanged
Label	Name Text	lblPrice Price:	N/A
Text Box	Name ReadOnly	txtPrice True	TxtPrice_TextChanged
Label	Name Text	lblStaffName Staff Name:	N/A
Combo Box	Name	cbStaffName	CbStaffName_TextChanged
Label	Name	lblDateCreated	N/A

	Text	Date Created:	
Date Time Picker	Name	dtpDateCreated	DtpDateCreated_ValueChanged
Button	Name Text Cursor FlatStyle	btnBooking Create Booking Hand Flat	BtnBooking_Click
Error Provider	Name	ep	N/A



- Update



**1) UcBooking:** This User Control will be displayed when user clicks the Booking button on the main menu and then the Update Booking button.

**2) – 24) Refer to page ~33~**

**25), 26) Search Text Box and Button:** Controls that can be used to search for a booking/s made by a client by inputting their surname and forename. The booking record/s should appear in the dgvBookings [27].

**27) dgvBookings:** Data Grid View that is displayed once the screen is loaded along with the search controls [25), 26)] so the user will be able to double click on a booking record which displays the rest of the controls to verify or update the booking.

**28) btnBack:** Once a booking record is selected from dgvBookings [27)] and it has disappeared, the user will be able to click this button to go back to the Data Grid View.

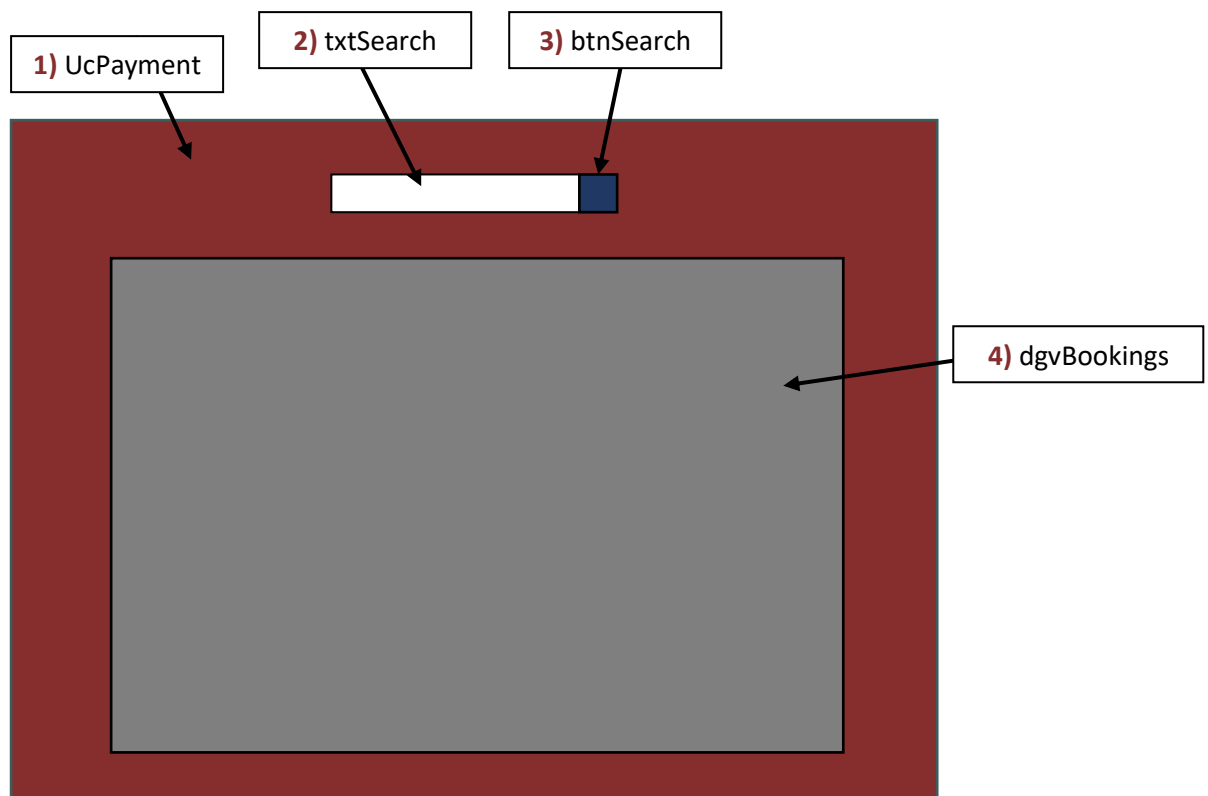
**29) btnDelete:** Button that will delete the current booking displayed from the database.

Control	Properties	Value	Events
User Control	Name Size	UcBooking 760 x 550 (px)	N/A
Label	Name Text	lblBookingNo BookingNo:	N/A
Text Box	Name ReadOnly	txtBookingNo True	TxtBookingNo_TextChanged
Label	Name Text	lblDogName Dog Name:	N/A
Combo Box	Name	cbDogName	CbDogName_TextChanged
Label	Name Text	lblClientName Client Name:	N/A
Combo Box	Name	cbClientName	CbClientName_TextChanged
Label	Name Text	lblDate Date:	N/A
Date Time Picker	Name Format	dtpDate Short	DtpDate_ValueChanged
Label	Name Text	lblStartTime Start Time:	N/A
Date Time Picker	Name CustomFormat	dtpStartTime 'hh:mm tt'	DtpStartTime_ValueChanged
Label	Name Text	lblServiceOption ServiceOption:	N/A
Combo Box	Name Items	cbServiceOption [Option 1, Option 2, Option 3]	CbServiceOption_TextChanged
Check Box	Name Text	chbExtraOption Add Extra Option	ChbExtraOption_CheckState Changed
Label	Name Text	lblDuration Duration:	N/A
Text Box	Name ReadOnly	txtDuration True	TxtDuration_TextChanged
Label	Name Text	lblPrice Price:	N/A
Text Box	Name ReadOnly	txtPrice True	TxtPrice_TextChanged
Label	Name Text	lblStaffName Staff Name:	N/A
Combo Box	Name	cbStaffName	CbStaffName_TextChanged
Label	Name Text	lblDateCreated Date Created:	N/A
Date Time Picker	Name	dtpDateCreated	DtpDateCreated_ValueChanged

Button	Name Text Cursor FlatStyle	btnBooking Update Booking Hand Flat	BtnBooking_Click
Error Provider	Name	ep	N/A
Text Box	Name AutoComplete Mode	txtSearch SuggestAppend	N/A
Button	Name Icon	btnSearch [Search Icon]	BtnSearch_Click
Data Grid View	Name Location Size	dgvBookings 35 x 95 (px) 690 x 415 (px)	DgvBookings_CellContent DoubleClick

## Payment

### - Search



- 1) **UcPayment:** This User Control will be displayed when user clicks the Payment button on the main menu and then the Search Booking button.
- 2), 3) **Search Text Box and Button:** Controls that can be used to search for a booking/s made by a client by inputting their surname and forename. The booking record/s should appear in the dgvBookings [4].
- 4) **dgvBookings:** Data Grid View that is displayed all the bookings in the database so the user will be able to double click on a booking record which displays transports the user to the View Payment Page which fills all the input fields with the payment details of the selected booking record to be updated by the user.

Control	Properties	Value	Events
User Control	Name Size	UcPayment 760 x 550 (px)	N/A
Text Box	Name AutoCompleteMode	txtSearch SuggestAppend	N/A
Button	Name Icon	btnSearch [Search Icon]	BtnSearch_Click
Data Grid View	Name Location Size	dgvBookings 35 x 95 (px) 690 x 415 (px)	DgvBookings_CellContent DoubleClick

- View

The diagram illustrates the 'UcPayment' User Control interface. It features a dark red background with white text and input fields. The components are numbered as follows:

- 1) UcPayment**: The main container for the payment form.
- 2) lblPaymentNo**: Label for the Payment Number.
- 3) txtPaymentNo**: Text box for the Payment Number.
- 4) lblBookingNo**: Label for the Booking Number.
- 5) txtBookingNo**: Text box for the Booking Number.
- 6) lblDate**: Label for the Date.
- 7) dtpDate**: Date Time Picker for the Date.
- 8) lblCost**: Label for the Cost.
- 9) txtCost**: Text box for the Cost.
- 10) lblPaymentMethod**: Label for the Payment Method.
- 11) cbPaymentMethod**: Combo Box for the Payment Method.
- 12) lblCardNumber**: Label for the Card Number.
- 13) txtCardNumber**: Text box for the Card Number.
- 14) lblSecurityCode**: Label for the Security Code.
- 15) txtSecurityCode**: Text box for the Security Code.
- 16) chbPaid**: Check Box for 'Paid'.
- 17) btnPayment**: Button labeled 'Create Payment'.
- 18) ep**: An error message icon (a red 'X') with the text 'ep'.

- 1) UcPayment:** This User Control will be displayed when user clicks the Payment button on the main menu and then the View Booking button or if they have double clicked a booking record on the Search Payment Page.
- 2), 4), 6), 8), 10), 12), 14) Labels:** Indicates what the text field adjacent is for.
- 3), 5) txtPaymentNo & txtBookingNo:** These text boxes should be read only and should automatically be set to the values of the current payment record shown.
- 7) dtpDate:** Date Time Picker to hold the date the payment was made/completed.
- 9) txtCost:** Text Box that should hold the total amount due left to be paid by the client. This should be automatically be calculated and set.
- 11) cbPaymentMethod:** Combo Box to hold the method of payment. If 'Debit Card' or 'Credit Card' is selected, then the Card Number and Security Code controls will be shown.
- 13), 15) txtCardNumber & txtSecurityCode:** These text boxes will be hidden from the user unless the payment method is changed to 'Debit Card' or 'Credit Card' in cbPaymentMethod [11]. These text boxes will hold the card details for a successful payment.
- 16) chbPaid:** This should be checked if the payment has been successfully received.
- 17) btnPayment:** Button that will validate and then update the existing payment record with all the data entered into the fields.

Control	Properties	Value	Events
User Control	Name Size	UcPayment 760 x 550 (px)	N/A
Label	Name Text	lblPaymentNo PaymentNo:	N/A
Text Box	Name ReadOnly	txtPaymentNo True	TxtPaymentNo_TextChanged
Label	Name Text	lblBookingNo BookingNo:	N/A
Text Box	Name ReadOnly	txtBookingNo True	TxtBookingNo_TextChanged
Label	Name Text	lblDate Date:	N/A
Date Time Picker	Name Format	dtpDate Short	DtpDate_ValueChanged
Label	Name Text	lblCost Cost:	N/A
Text Box	Name	txtCost	TxtCost_TextChanged
Label	Name Text	lblPaymentMethod Payment Method:	N/A
Combo Box	Name Items	cbPaymentMethod [Cash, Cheque, Debit Card, Credit Card]	CbPaymentMethod_TextChanged
Label	Name Text Visible	lblCardNumber Card Number: False	N/A
Text Box	Name Visible	txtCardNumber False	TxtCardNumber_TextChanged
Label	Name Text Visible	lblSecurityCode Security Code: False	N/A
Text Box	Name Visible	txtSecurityCode False	TxtSecurityCode_TextChanged
Check Box	Name Text	chbPaid Paid	ChbPaid_CheckStateChanged
Button	Name Text Cursor FlatStyle	btnPayment Create Payment Hand Flat	BtnPayment_Click
Error Provider	Name	ep	N/A





# TESTING



## Test Plan

### Main Menu

Test No.	Test Data/Reasons for Test	Expected Outcome	User Requirement
1	Test to see if the main menu page loads correctly	The main menu should load with the appropriate controls (e.g. Buttons, Labels, Picture Boxes etc.)	2.01
2	Test to ensure the Home button functions correctly.	The Home button, once clicked, should take the user to the Home Page.	2.01
3	Test to see if the Register button functions correctly.	The Register button, once clicked, should transfer the user to the Registration Page with the default start being the Client Registration.	2.01
4	Test to see if the Update button functions correctly.	The Update button, once clicked, should transfer the user to the Update Page with the default start being the Client Update.	2.01
5	Test to ensure the Booking button functions correctly.	The Booking button, once clicked, should transfer the user to the Booking Page with the default start being the Create Booking stage.	2.01
6	Test to check if the Payment button functions correctly.	The Payment button, once clicked, should transfer the user to the Payment Page with the default start being the Search Payment stage.	2.01
7	Test to check if the Reports button functions correctly.	The Reports button, once clicked, should take the user to the Reports Page.	2.01
8	Test to ensure the Title Label is being updated.	The Title Label should change its text accordingly with the last button clicked, e.g. when Register button is pressed then the Title Label should read 'Register'.	2.02
9	Test to check if the last clicked button is being highlighted.	The small rectangle panel should move to be on top of the last clicked button.	2.03
10	Test to ensure the tool bar is being displayed on top of each page.	The tool bar containing various buttons to change the current page when the Register, Update, Booking and Payment buttons are pressed.	2.04
11	Test to ensure the buttons on the tool bar are functioning correctly.	The buttons on the tool bar should change the current page within the current phase, e.g. within the Register phase, the tool bar should allow the	2.05

		user to switch between registering a Client and registering a Dog or a Staff member.	
12	Test to ensure the last clicked button on the tool bar is being highlighted.	The last button clicked on the tool bar should be coloured with the same colour as the page shown.	2.05
13	Test to ensure the Minimise button functions appropriately.	The Minimise button should restore the application to the desktop's taskbar when clicked.	2.06
14	Test to ensure the Exit button functions correctly.	The Close button, when clicked, should close the application fully. When the button is hovered over, it should turn red.	2.06

## Home

## Client

### - Register

Test No.	Test Data/Reasons for Test	Expected Outcome	User Requirement
15	Test to see if the Client Registration page loads appropriately.	The page should load in the Registration stage with the appropriate controls being shown.	4.01
16	Test to check if the next available ClientID is displayed.	When the page loads, the ClientID text box should be automatically filled with the new available ClientID and should be set to read only.	4.03
17	Test to check if the Register Client button on the page functions correctly.	The Register Client button, once clicked, should check if all the user input is valid and then register them or if invalid, place errors where necessary.	4.06
18	Test to see if the ClientID text box displays errors when necessary.	The ClientID text box should display errors if the text entered does not equal 4 digits.	4.02/4.07
19	Test to see if the Forename text box displays errors when necessary.	The Forename text box should show errors when the user leaves it empty or when they input digits or unsuitable punctuation or when they input over 30 characters.	4.02/4.08
20	Test to see if the Surname text box displays errors when necessary.	The Surname text box should show errors when the user leaves it empty or when they input digits or unsuitable punctuation or when they input over 30 characters.	4.02/4.08
21	Test to see if the Sex combo box shows errors when necessary.	The Sex combo box should be 'selection only' so the user cannot	4.02/4.09

		select anything invalid. Errors should flash if left empty.	
22	Test to check if the Date of Birth date time picker displays errors when necessary.	The Date of Birth should show errors if the client is under 18 years old and if the date entered is over 100 years in the past.	4.02/4.10
23	Test to ensure the Address text box flashes errors when necessary.	The Address text box should flash errors if the user enters anything other than letters and numbers and the value exceeds 50 characters. Errors should flash if left empty.	4.02/4.11
24	Test to ensure the City text box displays errors when necessary.	The City text box should flash errors if the user inputs anything other than letters and/or forward slash and if the value exceeds 50 characters. Errors should flash if left empty.	4.02/4.12
25	Test to check if the Postcode text box displays errors when needed.	The Postcode text box should show errors if the user inputs a postcode in an invalid format and if the value exceeds 8 characters. Errors should display if left empty.	4.02/4.13
26	Test to check if the E-mail text box displays errors when needed.	The E-mail text box should show errors if the user inputs an e-mail in an invalid format and if the value exceeds 50 characters. Errors should flash if left empty.	4.02/4.14
27	Test to ensure the Phone Number text box shows errors when deemed necessary.	The Phone Number text box should display errors if the user enters anything other than digits and if the phone number is below 11 characters and above 16. A plus ('+') is allowed at the start. Errors should display if left empty.	4.02/4.15
28	Test to ensure the Date Joined date time picker is set correctly.	The Date Joined date time picker should be automatically set to the current date when the new client is registered.	4.02/4.16
29	Test to see if the user is transferred to the Dog Registration after a successful registration.	After a successful Client Registration, the user should be transferred to the Dog Registration with the new client's name being automatically set to be the owner of the new dog.	4.17

- Update

Test No.	Test Data/Reasons for Test	Expected Outcome	User Requirement
29	Test to see if the Client Update page loads appropriately.	The page should load in the Update stage with the appropriate controls being shown.	4.01
30	Test to check if all the input fields are correctly filled.	When the page loads, all the input fields should automatically be filled with the details of the first client in the database.	4.04
31	Test to see if the Navigation buttons function acceptably.	The Previous and Next buttons should navigate through the different client details in the database in order of their ClientID.	4.04
32	Test to see if the Search controls function correctly.	The Search text box should suggest available clients when the user enters a name and the Search button, once clicked should automatically fill the fields with that client's data.	4.04
33	Test to see if the Delete Client button works as expected.	The Delete Client button should delete the current client shown and show a message box for the user to confirm the deletion and another if the user selects 'Yes' confirming a successful deletion.	4.05
34	Test to check if the Update Client button on the page functions correctly.	The Update Client button, once clicked, should check if all the user input is valid and update the client's details or if invalid, place errors where necessary.	4.06
35	Test to see if the ClientID text box displays errors when necessary.	The ClientID text box should display errors if the text entered does not equal 4 digits	4.02/4.07
36	Test to see if the Forename text box displays errors when necessary.	The Forename text box should show errors when the user leaves it empty or when they input digits or unsuitable punctuation or when they input over 30 characters.	4.02/4.08
37	Test to see if the Surname text box displays errors when necessary.	The Surname text box should show errors when the user leaves it empty or when they input digits or unsuitable punctuation or when they input over 30 characters.	4.02/4.08
38	Test to see if the Sex combo box shows errors when necessary.	The Sex combo box should be 'selection only' so the user cannot	4.02/4.09

		select anything invalid. Errors should flash if left empty.	
39	Test to check if the Date of Birth date time picker displays errors when necessary.	The Date of Birth should show errors if the client is under 18 years old and if the date entered is over 100 years in the past.	4.02/4.10
40	Test to ensure the Address text box flashes errors when necessary.	The Address text box should flash errors if the user enters anything other than letters and numbers and the value exceeds 50 characters. Errors should flash if left empty.	4.02/4.11
41	Test to ensure the City text box displays errors when necessary.	The City text box should flash errors if the user inputs anything other than letters and/or forward slash and if the value exceeds 50 characters. Errors should flash if left empty.	4.02/4.12
42	Test to check if the Postcode text box displays errors when needed.	The Postcode text box should show errors if the user inputs a postcode in an invalid format and if the value exceeds 8 characters. Errors should display if left empty.	4.02/4.13
43	Test to check if the E-mail text box displays errors when needed.	The E-mail text box should show errors if the user inputs an e-mail in an invalid format and if the value exceeds 50 characters. Errors should flash if left empty.	4.02/4.14
44	Test to ensure the Phone Number text box shows errors when deemed necessary.	The Phone Number text box should display errors if the user enters anything other than digits and if the phone number is below 11 characters and above 16. A plus ('+') is allowed at the start. Errors should display if left empty.	4.02/4.15
45	Test to ensure the Date Joined date time shows errors when needed.	The Date Joined date time picker should show errors when the user sets it to a date in the future.	4.02/4.16
46	Test to check if the View Dogs Owned button functions correctly.	The View Dogs Owned button, when clicked, should show a data grid view displaying all the dogs owned by the current client	4.18

## Dog

### - Register

Test No.	Test Data/Reasons for Test	Expected Outcome	User Requirement
47	Test to see if the Dog Registration page loads appropriately.	The page should load in the Registration stage with the appropriate controls being shown.	6.01
48	Test to check if the next available DogID is displayed.	When the page loads, the DogID text box should be automatically filled with the new available DogID and should be set to read only.	6.03
49	Test to check if the Register Dog button on the page functions correctly.	The Register Dog button, once clicked, should check if all the user input is valid and then register them or if invalid, place errors where necessary.	6.07
50	Test to see if the DogID text box displays errors when necessary.	The DogID text box should display errors if the text entered does not equal 4 digits.	6.02/6.08
51	Test to see if the Client Name combo box displays errors when necessary.	The Client Name combo box should be 'selection only' so the user cannot select anything invalid. Errors should flash if left empty.	6.02/6.09
52	Test to see if the Name text box displays errors when necessary.	The Name text box should show errors when the user leaves it empty or anything other than letters is inputted or when they input over 30 characters.	6.02/6.10
53	Test to check if the Breed text box shows errors when needed.	The Breed text box should show errors when the user leaves it empty or anything other than letters is inputted or when they input over 30 characters.	6.02/6.11
54	Test to see if the Sex combo box shows errors when necessary.	The Sex combo box should be 'selection only' so the user cannot select anything invalid. Errors should flash if left empty.	6.02/6.12
55	Test to check if the Date of Birth date time picker displays errors when necessary.	The Date of Birth should show errors if the dog is over 50 years old and if the date entered is in the future.	6.02/6.13
56	Test to ensure the Medical Notes rich text box is set correctly.	The Medical Notes rich text box should be automatically set to 'None' when the new dog is registered.	6.02/6.15



- Update

Test No.	Test Data/Reasons for Test	Expected Outcome	User Requirement
57	Test to see if the Dog Update page loads appropriately.	The page should load in the Update phase with the appropriate controls being shown.	6.01
58	Test to check if all the input fields are correctly filled.	When the page loads, all the input fields should automatically be filled with the details of the first dog in the database.	6.04
59	Test to see if the Navigation buttons function acceptably.	The Previous and Next buttons should navigate through the different dog details in the database in order of their DogID.	6.04
60	Test to see if the Search controls function correctly.	The Search text box should suggest available dogs when the user enters a their name along with their owner's name and the Search button, when clicked should automatically fill the fields with that dog's data.	6.04
61	Test to see if the Delete Dog button works as expected.	The Delete Dog button should delete the current dog shown and show a message box for the user to confirm the deletion and another if the user selects 'Yes' confirming a successful deletion.	6.05
62	Test to ensure the dog's profile picture is being loaded correctly.	When a new dog is loaded onto the page, their profile picture should also show to quickly identify them.	6.06
63	Test to check if the Update Dog button on the page functions correctly.	The Update Dog button, once clicked, should check if all the user input is valid and then update their details or if invalid, place errors where necessary.	6.07
64	Test to see if the DogID text box displays errors when necessary.	The DogID text box should display errors if the text entered does not equal 4 digits.	6.02/6.08
65	Test to see if the Client Name combo box displays errors when necessary.	The Client Name combo box should be 'selection only' so the user cannot select anything invalid. Errors should flash if left empty.	6.02/6.09
66	Test to see if the Name text box displays errors when necessary.	The Name text box should show errors when the user leaves it empty or anything other than letters is inputted or when they input over 30 characters.	6.02/6.10

67	Test to check if the Breed text box shows errors when needed.	The Breed text box should show errors when the user leaves it empty or anything other than letters is inputted or when they input over 30 characters.	6.02/6.11
68	Test to see if the Sex combo box shows errors when necessary.	The Sex combo box should be 'selection only' so the user cannot select anything invalid. Errors should flash if left empty.	6.02/6.12
69	Test to check if the Date of Birth date time picker displays errors when necessary.	The Date of Birth should show errors if the dog is over 50 years old and if the date entered is in the future.	6.02/6.13
70	Test to check if the View Medical Notes button functions correctly.	The View Medical Notes button, once clicked, should hide the dog profile picture box and show the Medical Notes rich text box and label. The opposite should occur when clicked again.	6.15
71	Test to ensure the Medical Notes rich text box displays errors when necessary	The Medical Notes rich text box should show errors if left empty or if over 100 characters is entered.	6.02/6.15
72	Test to ensure the View Vaccination History button functions correctly.	The View Vaccination History button should show the data grid view showing a list of all the vaccinations received by the dog and when.	6.16

## Staff

### - Register

Test No.	Test Data/Reasons for Test	Expected Outcome	User Requirement
73	Test to see if the Staff Registration page loads appropriately.	The page should load in the Registration stage with the appropriate controls being shown.	6.01
74	Test to check if the next available StaffID is displayed.	When the page loads, the StaffID text box should be automatically filled with the new available StaffID and should be set to read only.	6.03
75	Test to check if the Register Staff button on the page functions correctly.	The Register Staff button, once clicked, should check if all the user input is valid and then register them or if invalid, place errors where necessary.	6.06
76	Test to see if the StaffID text box displays errors when necessary.	The StaffID text box should display errors if the text entered does not equal 4 digits.	6.02/6.07
77	Test to see if the Forename text box displays errors when necessary.	The Forename text box should show errors when the user leaves it empty or when they input digits or unsuitable punctuation or when they input over 30 characters.	6.02/6.08
78	Test to see if the Surname text box displays errors when necessary.	The Surname text box should show errors when the user leaves it empty or when they input digits or unsuitable punctuation or when they input over 30 characters.	6.02/6.08
79	Test to see if the Sex combo box shows errors when necessary.	The Sex combo box should be 'selection only' so the user cannot select anything invalid. Errors should flash if left empty.	6.02/6.09
80	Test to check if the Date of Birth date time picker displays errors when necessary.	The Date of Birth should show errors if the staff member is under 16 years old and if the date entered is over 70 years in the past.	6.02/6.10
81	Test to ensure the Address text box flashes errors when necessary.	The Address text box should flash errors if the user enters anything other than letters and numbers and the value exceeds 50 characters. Errors should flash if left empty.	6.02/6.11
82	Test to ensure the City text box displays errors when necessary.	The City text box should flash errors if the user inputs anything other than letters and/or forward slash and if the	6.02/6.12

		value exceeds 50 characters. Errors should flash if left empty.	
83	Test to check if the Postcode text box displays errors when needed.	The Postcode text box should show errors if the user inputs a postcode in an invalid format and if the value exceeds 8 characters. Errors should display if left empty.	6.02/6.13
84	Test to check if the E-mail text box displays errors when needed.	The E-mail text box should show errors if the user inputs an e-mail in an invalid format and if the value exceeds 50 characters. Errors should flash if left empty.	6.02/6.14
85	Test to ensure the Phone Number text box shows errors when deemed necessary.	The Phone Number text box should display errors if the user enters anything other than digits and if the phone number is below 11 characters and above 16. A plus ('+') is allowed at the start. Errors should display if left empty.	6.02/6.15
86	Test to ensure the Job Contract check boxes shows errors when needed.	The user should show errors if the user does not pick on the job contract check boxes.	6.02/6.16
87	Test to see if the Job Position text box displays errors when needed.	The Job Position combo box should be 'selection only' so the user cannot select anything invalid. Errors should flash if left empty.	6.02/6.17
88	Test to check if the Salary numeric up-down shows errors when appropriate.	The Salary numeric up-down should show be set to 2 decimal places by default and should show an error when the user enters a salary over 10 digits in length.	6.02/6.18
89	Test to ensure the Date Joined date time picker is set correctly.	The Date Joined date time picker should be automatically set to the current date when the new staff member is registered.	6.02/6.19

- Update

Test No.	Test Data/Reasons for Test	Expected Outcome	User Requirement
90	Test to see if the Staff Update page loads appropriately.	The page should load in the Update phase with the appropriate controls being shown.	6.01
74	Test to check if the next available StaffID is displayed.	When the page loads, the StaffID text box should be automatically filled with the new available StaffID and should be set to read only.	6.03
75	Test to check if the Register Staff button on the page functions correctly.	The Register Staff button, once clicked, should check if all the user input is valid and then register them or if invalid, place errors where necessary.	6.06
76	Test to see if the StaffID text box displays errors when necessary.	The StaffID text box should display errors if the text entered does not equal 4 digits.	6.02/6.07
77	Test to see if the Forename text box displays errors when necessary.	The Forename text box should show errors when the user leaves it empty or when they input digits or unsuitable punctuation or when they input over 30 characters.	6.02/6.08
78	Test to see if the Surname text box displays errors when necessary.	The Surname text box should show errors when the user leaves it empty or when they input digits or unsuitable punctuation or when they input over 30 characters.	6.02/6.08
79	Test to see if the Sex combo box shows errors when necessary.	The Sex combo box should be 'selection only' so the user cannot select anything invalid. Errors should flash if left empty.	6.02/6.09
80	Test to check if the Date of Birth date time picker displays errors when necessary.	The Date of Birth should show errors if the staff member is under 16 years old and if the date entered is over 70 years in the past.	6.02/6.10
81	Test to ensure the Address text box flashes errors when necessary.	The Address text box should flash errors if the user enters anything other than letters and numbers and the value exceeds 50 characters. Errors should flash if left empty.	6.02/6.11
82	Test to ensure the City text box displays errors when necessary.	The City text box should flash errors if the user inputs anything other than letters and/or forward slash and if the value exceeds 50 characters. Errors should flash if left empty.	6.02/6.12

83	Test to check if the Postcode text box displays errors when needed.	The Postcode text box should show errors if the user inputs a postcode in an invalid format and if the value exceeds 8 characters. Errors should display if left empty.	6.02/6.13
84	Test to check if the E-mail text box displays errors when needed.	The E-mail text box should show errors if the user inputs an e-mail in an invalid format and if the value exceeds 50 characters. Errors should flash if left empty.	6.02/6.14
85	Test to ensure the Phone Number text box shows errors when deemed necessary.	The Phone Number text box should display errors if the user enters anything other than digits and if the phone number is below 11 characters and above 16. A plus ('+') is allowed at the start. Errors should display if left empty.	6.02/6.15
86	Test to ensure the Job Contract check boxes shows errors when needed.	The user should show errors if the user does not pick on the job contract check boxes.	6.02/6.16
87	Test to see if the Job Position text box displays errors when needed.	The Job Position combo box should be 'selection only' so the user cannot select anything invalid. Errors should flash if left empty.	6.02/6.17
88	Test to check if the Salary numeric up-down shows errors when appropriate.	The Salary numeric up-down should show be set to 2 decimal places by default and should show an error when the user enters a salary over 10 digits in length.	6.02/6.18
89	Test to ensure the Date Joined date time picker is set correctly.	The Date Joined date time picker should be automatically set to the current date when the new staff member is registered.	6.02/6.19
29	Test to see if the user is transferred to the Dog Registration after a successful registration.	After a successful Client Registration, the user should be transferred to the Dog Registration with the new client's name being automatically set to be the owner of the new dog.	6.17

## **Booking**

- Create

- Update



## **Payment**

- Search

- View



Testing

Main Menu

Home

**Client**

- Register

- Update

**Dog**

- Register

- Update



## **Staff**

- **Register**

- Update

## **Booking**

- Create

- Update

**Payment**

- **Search**

- View

**Payment**

- **Search**



# Evaluation







# Appendix