|  |
| --- |
|  |
| Code Complexities |
| JD Dog Care Training Subsystem |
|  |
|  |
|  |

|  |
| --- |
|  |

Contents

[Complexity 1 2](#_Toc69989058)

[Adding duplicate dog name to a customer 2](#_Toc69989059)

[The Code 3](#_Toc69989060)

[Complexity 2 4](#_Toc69989061)

[Ensuring that the date of birth is valid 4](#_Toc69989062)

[The Code 5](#_Toc69989063)

[Complexity 3 6](#_Toc69989064)

[DataGrid that updates each time a booking is made, displays how many places are remaining 6](#_Toc69989065)

[The Code 7](#_Toc69989066)

[Complexity 4 8](#_Toc69989067)

[When editing a booking, only relevant dogs and classes are available 8](#_Toc69989068)

[The Code 10](#_Toc69989069)

[Complexity 5 12](#_Toc69989070)

[When a dog is archived, all associated bookings are also archived or deleted. 12](#_Toc69989071)

[The Code 13](#_Toc69989072)

[Complexity 6 14](#_Toc69989073)

[If a booking is in the future, it is deleted instead of being archived 14](#_Toc69989074)

[The Code 15](#_Toc69989075)

[Complexity 7 16](#_Toc69989076)

[When editing a booking, duplicate bookings can’t be created 16](#_Toc69989077)

[The Code 17](#_Toc69989078)

[Complexity 8 18](#_Toc69989079)

[Customer Loyalty Discount 18](#_Toc69989080)

[The Code 20](#_Toc69989081)

[Complexity 9 22](#_Toc69989082)

[Bookings automatically archive when the associated class start date has passed, class also can’t be booked and won’t show in active bookings data grid 22](#_Toc69989083)

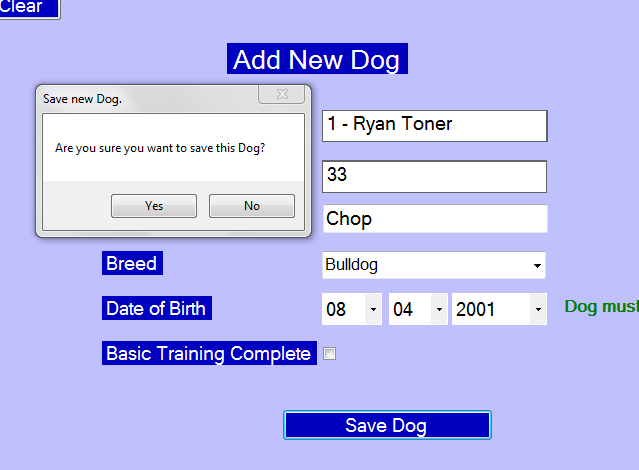
# Complexity 1

## Adding duplicate dog name to a customer

When adding Dogs to the system, if a dog with a duplicate name is added to a customer, the user is informed that a dog with this name already has been added to this customer, from here they are able to confirm if they want to add the dog anyway.

Although it is unlikely, a customer could have 2 dogs with the same name and should still be able to make bookings for both dogs regardless of the names being the same.

For example, This customer ‘Ryan Toner’ already has a dog named ‘Chop’ saved under his name in the system, trying to add another dog by the name of Chop will show the following:

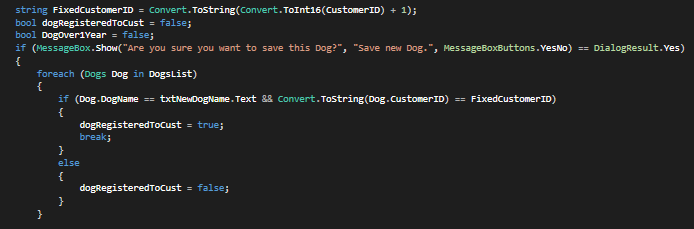
Upon clicking yes, the user will then be informed that a dog with this name already has been added to this customer.

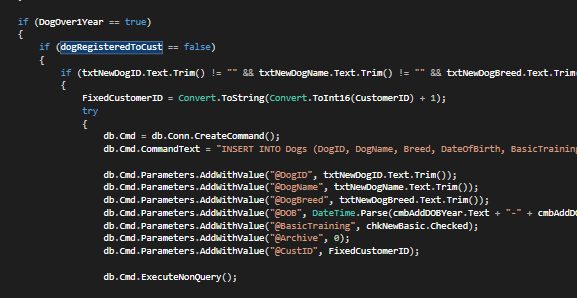


At this point the customer is aware the name is duplicate and can then make the decision to save anyway or cancel adding the new dog.

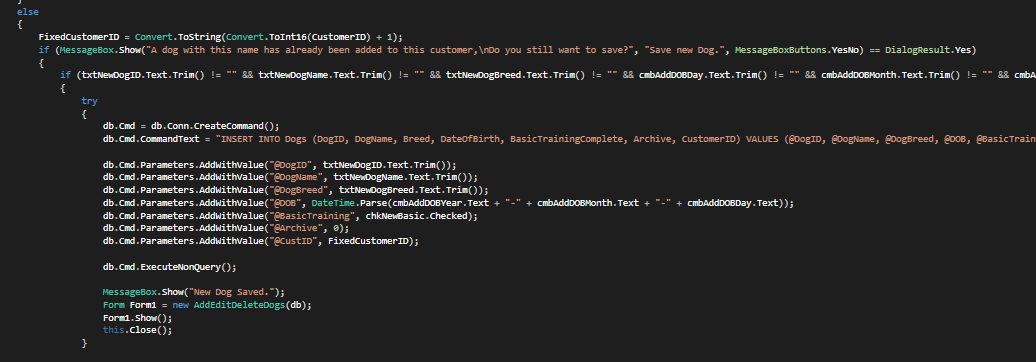
## The Code

Here is the code that checks each dog in the system to see if there is a dog with the same name already saved to this customer, if this is the case then the dogRegisteredToCust flag is set to true.





Later when the dog is actually being saved this flag is used to see if the dog should be saved right away or not.

Above is the code where the user is informed that a dog with this name has already been saved to this customer and is asked to confirm if they want to continue with the save anyway. If they continue with the save then the dog is added to the database.

# Complexity 2

## Ensuring that the date of birth is valid

When adding a new dog to the database, the customer must enter the date of birth using 3 combo boxes, this means that the customer is able to select invalid dates such as 31/02. If this is the case the system will automatically change these dates to become valid.

For example if the customer selects 31/02 it will automatically be changed to 28/02. However if the customer selects 29/02 the date will not be adjusted as this is a possible date.



Here a valid date has been entered but when the month is changed to 02 the day will be changed to 28.



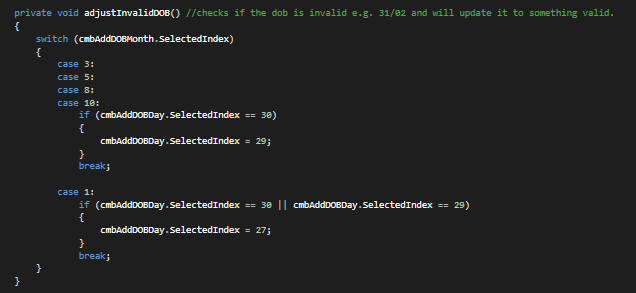
Now that the month has been changed to 02, the day has been updated to ensure that the date is valid. The user is now unable to select 30 or 31 as these dates are invalid.



29 can still be selected though as this is a possible date.

## The Code

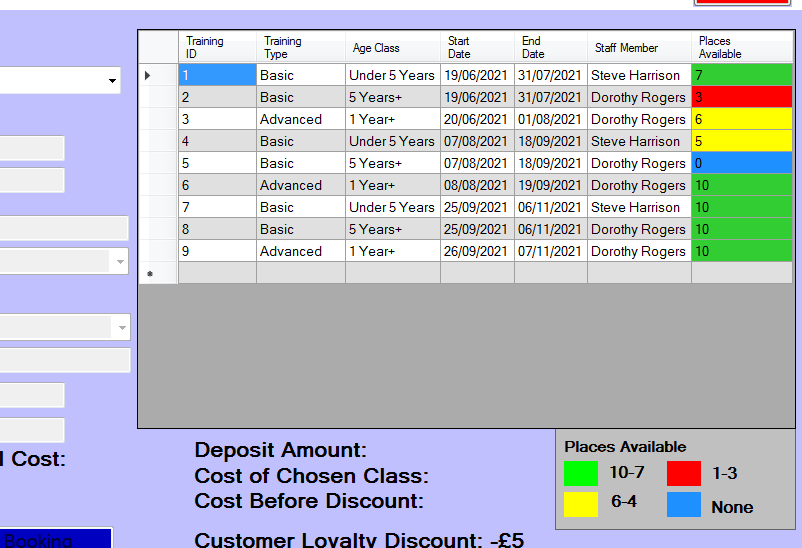
This method is called each time either the combo box for month or day are changed



This method checks to see if the month is one that only has 30 days in it, these being April, June, September and November, if this is the case then the day combo box is set to show 30 rather than 31. Additionally, if the month is February, if the month is set as 30 or 31, the month is then set as 28. In the code the numbers are reduced by 1 as this is just how counting works in c#.

# Complexity 3

## DataGrid that updates each time a booking is made, displays how many places are remaining



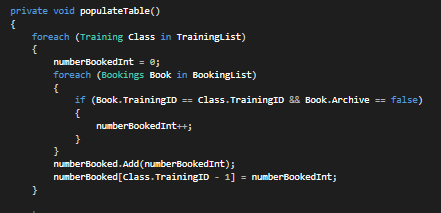
This DataGrid shows the number of places remaining in each currently active training class (they are only currently active if their start date is in the future). The amount of places remaining also coordinates to a specific colour which is explained below the DataGrid.



A booking was added to the first class, this means that the places available has gone down by 1 and has also turned yellow to show this change. Classes with 6-4 places are shown in Yellow.

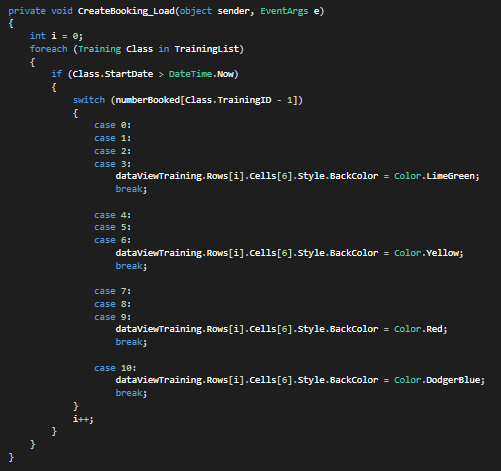
## The Code

This code is found at the beginning of the PopulateTable method which fills the datagrid with the information about active training classes.



This code calculates how many times each class appears in bookings to figure out how many places are still available.

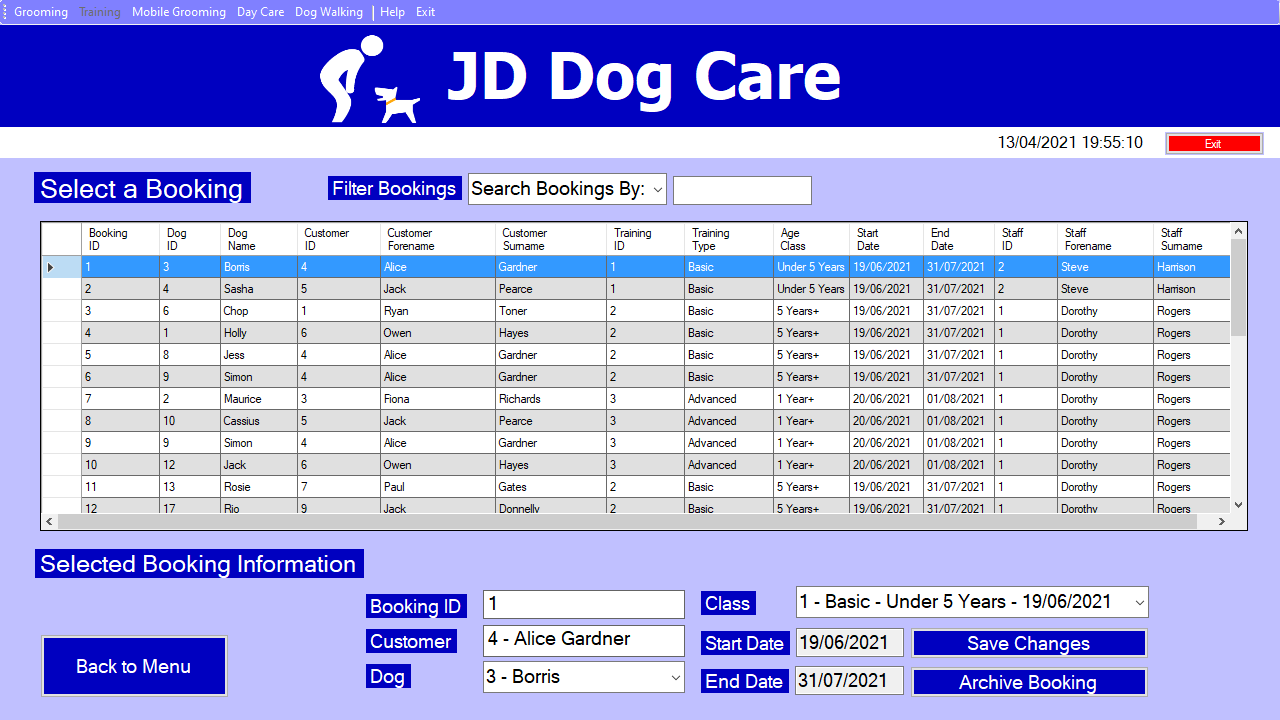
Below is the code that changes the colour of the table cells based on how many places are still available for that particular class.

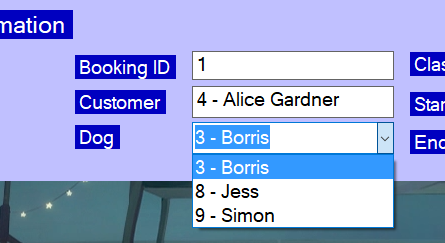


# Complexity 4

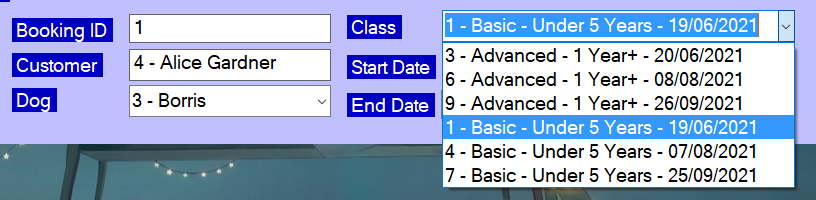
## When editing a booking, only relevant dogs and classes are available

This ensures that there are no bookings created that should not be possible to make such as a dog that is not eligible for advanced training being booked onto an advanced class. When a booking is selected, the only dogs that can be edited into the booking belong to the customer associated with the booking, the classes the booking can be changed to will be available based on the dogs age and training level.

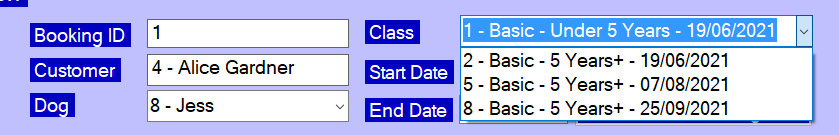


As shown here, booking 1 has been selected, Customer 4 Alice Gardner is the customer associated with this booking. Alice Gardner’s dogs are Borris, Jess and Simon so these are the only Dogs you are able to select when editing.

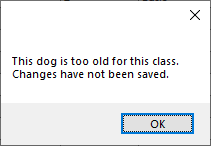
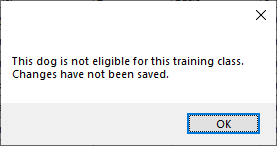
Additionally, only classes available for the dog will show in the Class drop-down list.



Boris is under 5 years and is eligible for advanced classes, the available classes reflect this.

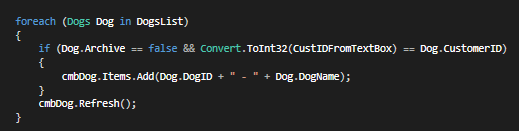


As Jess is over 5 years and not eligible for advanced training the available classes have changed. If the user tries to save when an invalid class is selected as shown in the above screenshot the following message will appear.

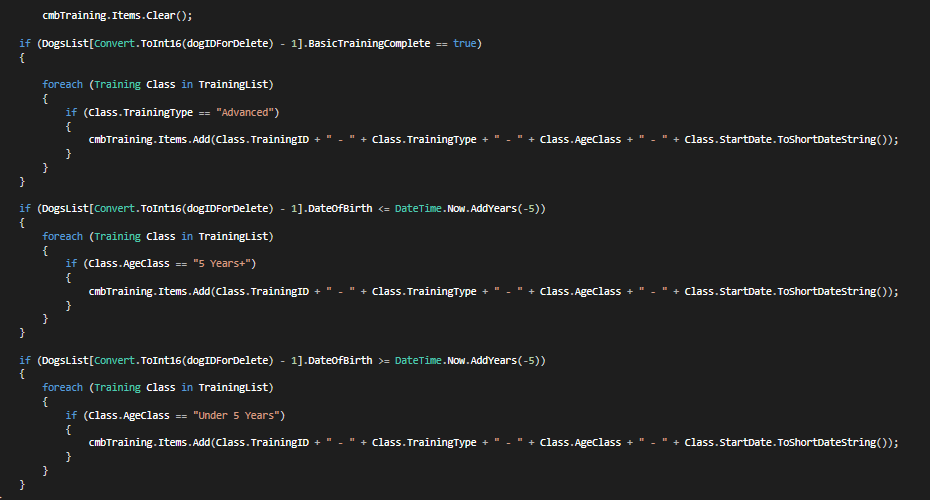
   
This depends on which aspect of the class makes it invalid for that dog.

## The Code

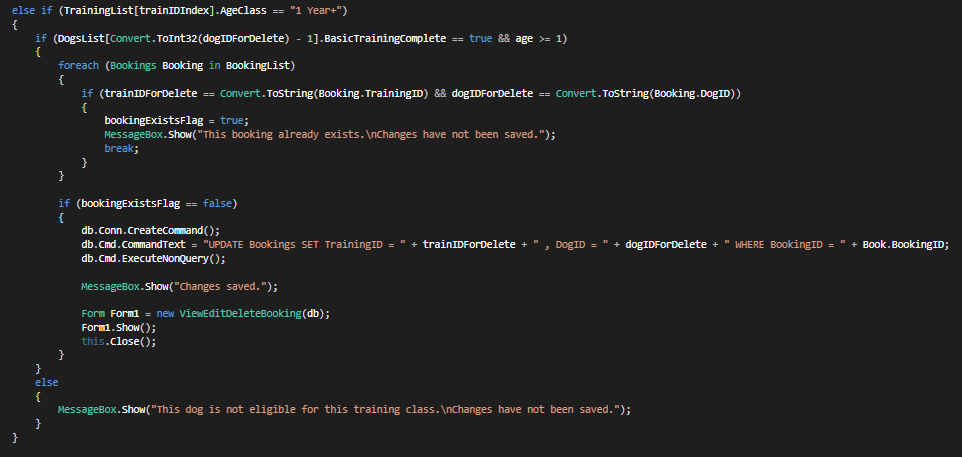
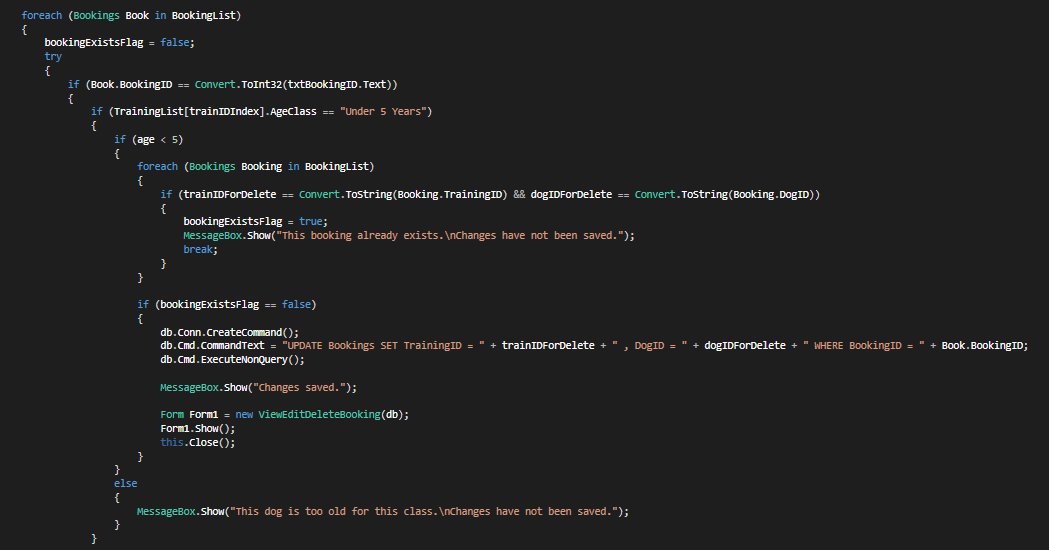
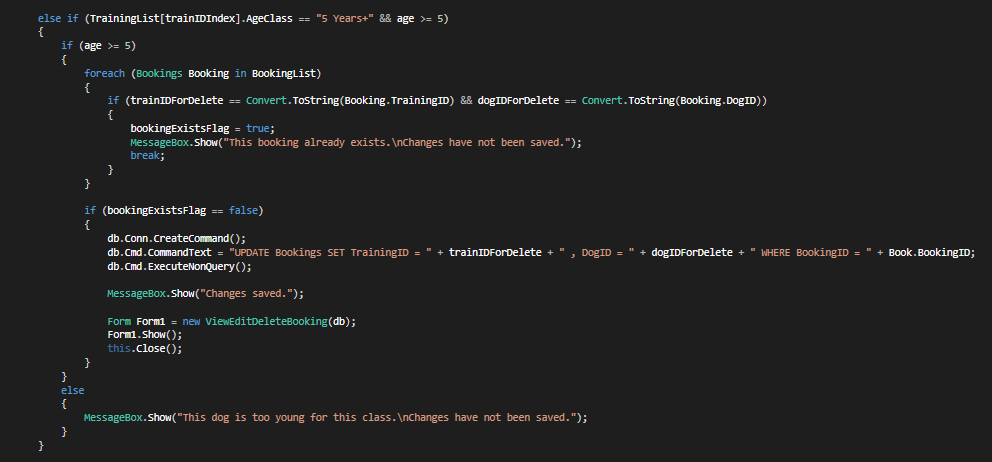
This code is what populates the Dog combo box with dogs registered to the customer associated with the booking. This will be executed when a cell on the data grid is selected.



Below is the code the populates the Class drop down list with only classes that are available for the selected dog. This is in the selected index changed event for the dog drop down list.



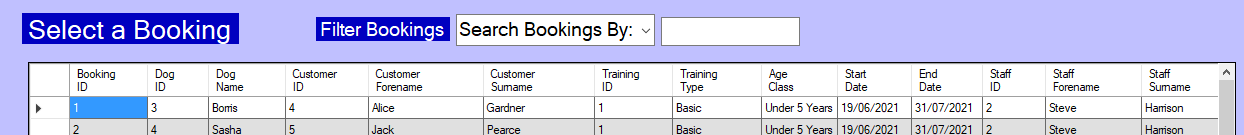
This code validates the booking, ensuring that it is not duplicate and that the selected dog is the right age for the selected class and that it is of the right training level, this is executed when the save changes button is clicked.



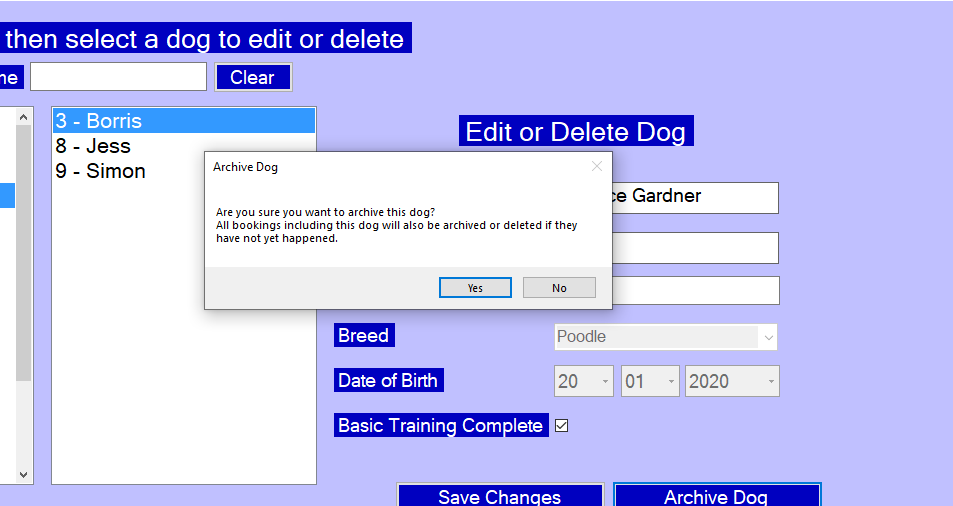
# Complexity 5

## When a dog is archived, all associated bookings are also archived or deleted.

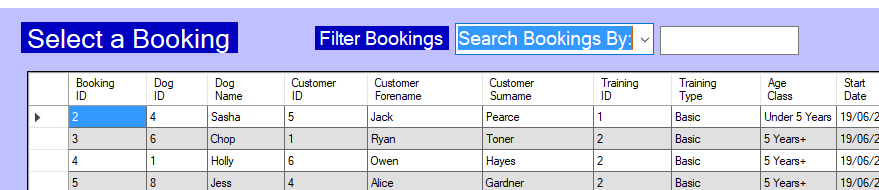
When a dog in the database is archived, all of its associated bookings are archived, as you can see from the screenshot below Booking ID 1 has been made for the dog Borris



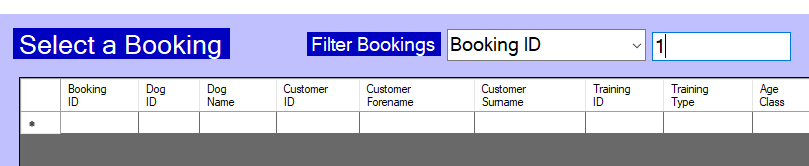
Boris is now going to be archived, the user is informed that all bookings associated with this dog are also going to be archived.



Now you can see that Booking ID 1 no longer appears in the “View/Edit/Delete Bookings” section of the system

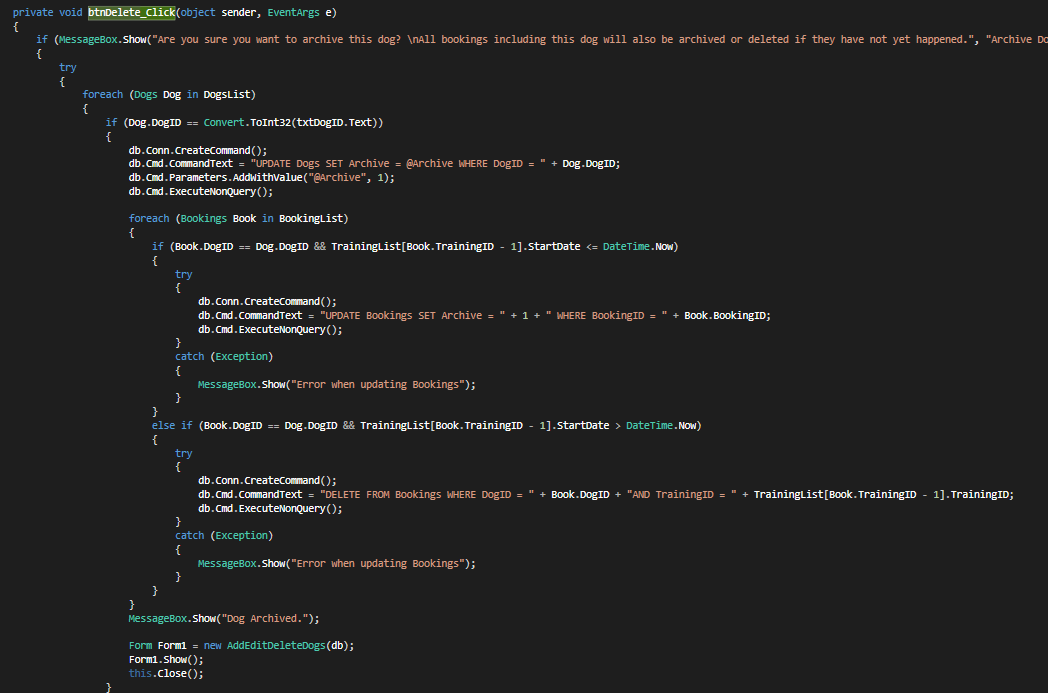


The bookings are even filtered here just to prove that the record is no longer active.



## The Code

Here is the code where the dog itself is archived, just below where this happens you can see the system going through each booking and archiving every booking with the associated dog id.



Archived dogs or bookings are not deleted from the database, instead they just can’t be edited or accessed while using the system. This is so that the JD Dog Care can keep track of their previous business. If the booking has not yet happened though it is fully deleted from the database. If the booking has already happened it is archived, whereas if it is in the future and hasn’t happened yet it is fully deleted from the database.

In this case the booking was fully deleted as it has not happened yet.



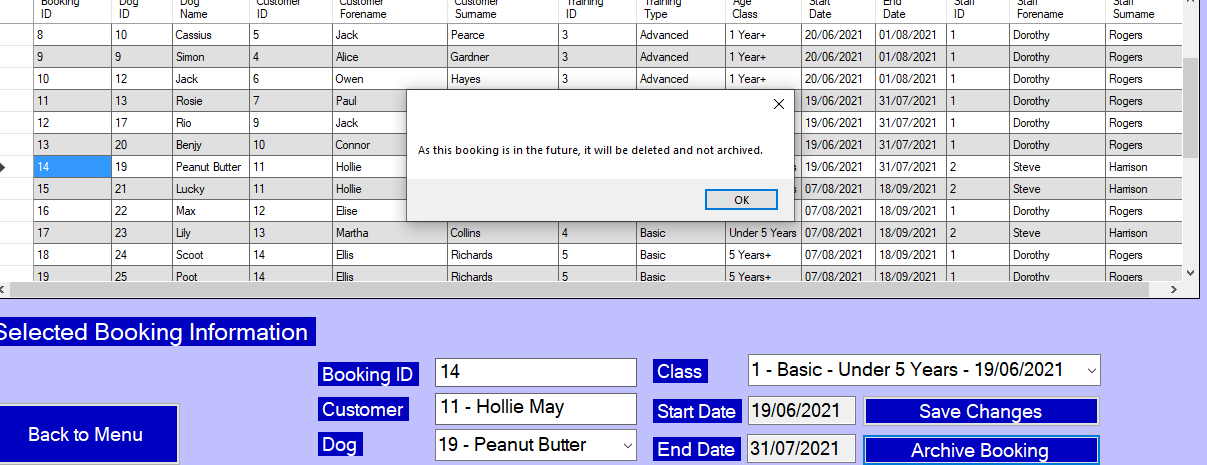
Here you can see Booking ID 1 is no longer in the database.

# Complexity 6

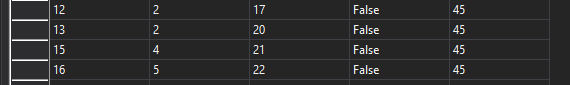
## If a booking is in the future, it is deleted instead of being archived

The reason for this is because the booking has not yet happened therefore there is no reason for JD Dog Care to save it, the customer will not have paid at this stage.

When clicking the archive button the user is informed that the booking is in the future and will be deleted because of this.



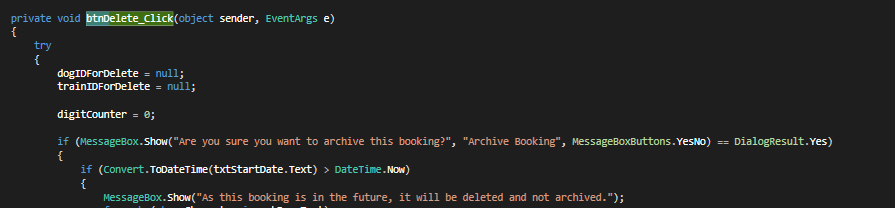
Now when navigating to the actual table data in the database.mdf file we can see that Booking ID 14 has been deleted from the system.



This will also happen when a dog is archived and all associated bookings are deleted or archived as shown in the complexity above.

## The Code

Below is the code where the user is asked to confirm that they want to archive/delete the booking, if the booking occurs in the future then it is deleted from the system fully. Whereas if it has already happened it is instead archived.

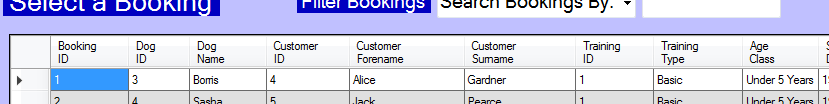


# Complexity 7

## When editing a booking, duplicate bookings can’t be created

When editing a booking the user is ab le to change the dog and class associated with a booking, but the user should not be able to change the dog and class to create a booking that has already been made before.

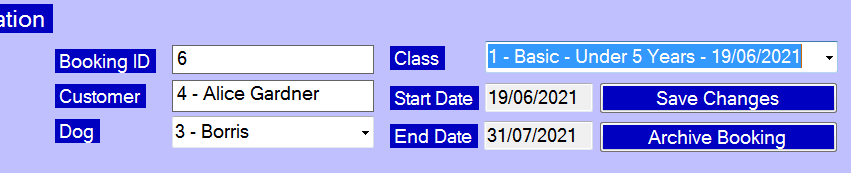
Below we can see that Booking ID 1 is linked to Customer ID 4 - Alice Gardner and Dog ID 3 - Borris. This booking has been made for Class ID 1.



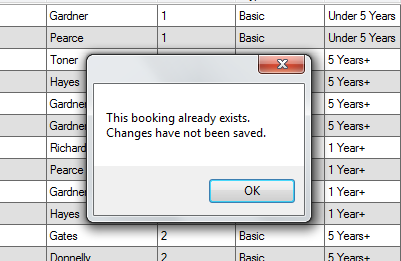
If we find another booking that is also linked to Customer ID 4 – Alice Gardner we can try to make it identical to Booking ID 1. Here you can see Booking ID 6 is also linked to Customer ID 4 – Alice Gardner.



Now the information about this booking will be changed so that it is the same as Booking ID 1.

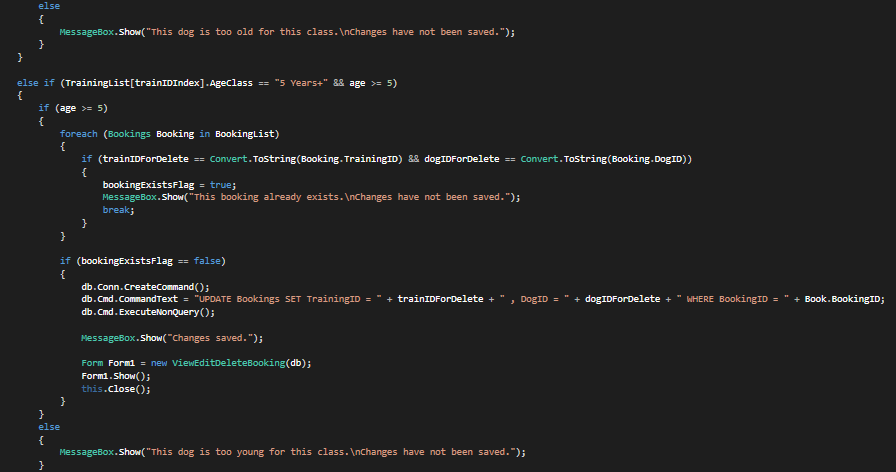


Upon clicking save the user is prompted with the following message:



The user is not able to create duplicate bookings using the edit bookings feature of the system.

## The Code



In the above code at the highlighted points you can see where the system searches through every booking to see if one already exists with the currently selected information, if this is the case the user is prompted with a messagebox, if not then the changes are saved.

There are 2 places where this occurs as the code changes slightly if the dog is over 5 years and above or under 5 years.

# Complexity 8

## Customer Loyalty Discount

If a customer has previously made a booking with JD Dog Care, then they receive a loyalty discount of £5 on all future bookings. This should encourage customers to make more bookings with JD Dog Care and in turn generate more money for the business.



The screenshot above shows a booking being created for Customer ID 1 – Ryan Toner who has previously made a booking with JD Dog Care, this means the £5 Customer loyalty discount is applied.

Meanwhile Customer ID 20 – Amy Peters has not made a booking with JD Dog Care before, so the discount is not applied.

These discounts are actually added to the BookingCost value of each booking as shown below.

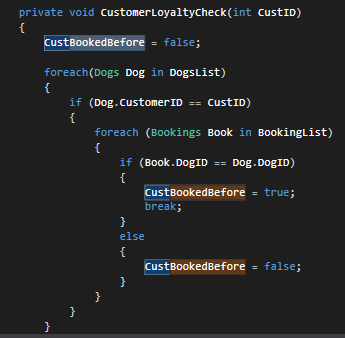




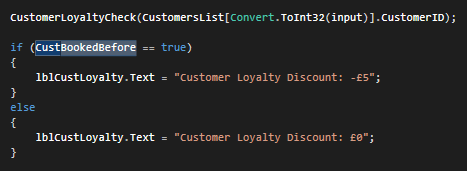
These records show the 2 bookings made with Ryan Toner and Amy Peters. Booking ID 30 is Ryan Toner’s booking, you can see that the booking cost is £55 rather than £60 as this discount has been applied whereas with Booking ID 31 – the one made with Amy Peters, the cost of the booking is still £60 as no discount has been applied.

## The Code

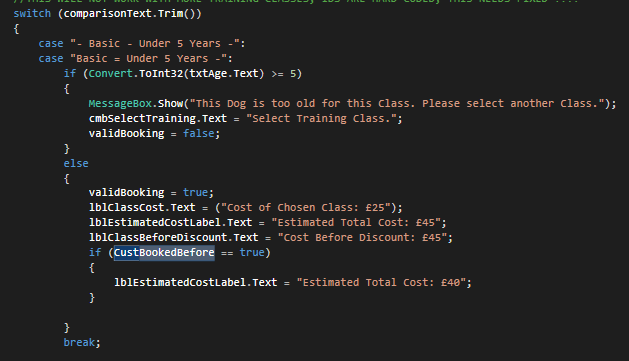
Here is the code that will check if the customer has previously made a booking before



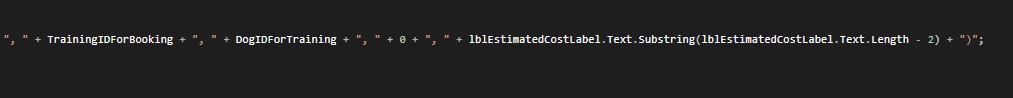
Below is where the user is informed that this customer is receiving a loyalty discount.



This is where the total estimated cost is calculated and shown to the user



Finally, the cost is added to the booking record along with the other pieces of information.

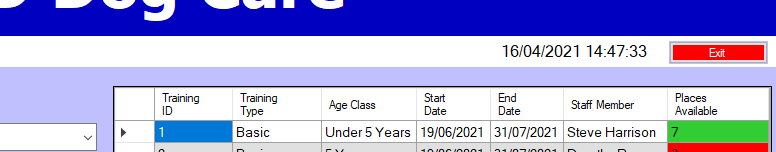


Here is a section of the SQL statement used when adding the booking record, the text shown on the estimated total cost label is what is used for the booking cost. Only the last characters are needed as these will be the numbers showing the cost. It is not possible for the booking cost to be anything other than 40, 45, 55 or 60 so 3+ digits are not necessary.

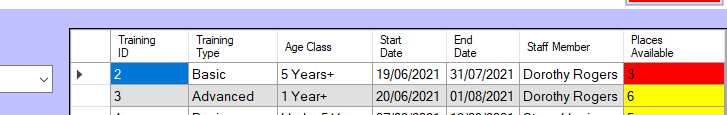
# Complexity 9

## Bookings automatically archive when the associated class start date has passed, class also can’t be booked and won’t show in active bookings data grid

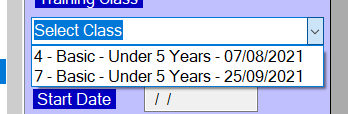
Firstly we can see that the Start Date for Training ID 1 is 19/06/2021, once we change this date to a date in the past the class will no longer show on the data grid and no longer be booked onto using the Create Booking form.



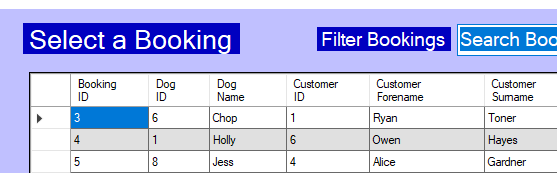
Now the start date of this booking will be edited to be 19/02/2021 which is a date that has already passed at the time of this test.



After changing the start date of Training ID 1, it no longer appears in the data grid showing currently active bookings, additionally it will not appear in the drop-down list where Classes can be selected for bookings to be created.

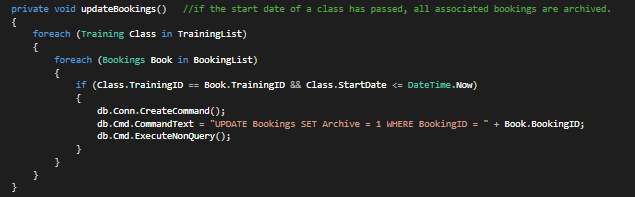


Now navigating to the View/Edit/Delete Bookings Form should show that there are some bookings not showing, these missing bookings are the ones archived that were booked onto this class.



The first booking showing is Booking ID 3, even though Booking ID 1 and 2 were booked onto the past class. This means that the bookings have been archived as the class has passed.

## The Code



All bookings are checked and if the start date of the associated class has passed, the booking is archived.