Huw Talliss

Data Structures and Variables

Variables

The Regex class represents a read-only regular expression. It also contains static methods that allow use of other regular expression classes without explicitly creating instances of the other classes.

Variable Name	Data Type	Scope	Location	Purpose
SeatID	String	Local	DataEntry	To hold and to display the Seat that is being booked
Day	Integer	Local	DataEntry	To hold and to display the day that the current seat is being booked for
Price	Float	Local	DataEntry	To hold the price of the current seat
Disabled	Boolean	Local	DataEntry	To hold whether the seat is a disabled seat or not
FirstNameRegex	Regex	Local	DataEntry	Used to check whether user entered data is valid
LastNameRegex	Regex	Local	DataEntry	Used to check whether user entered data is valid
HouseNameNumRegex	Regex	Local	DataEntry	Used to check whether user entered data is valid
Address1Regex	Regex	Local	DataEntry	Used to check whether user entered data is valid
Address2Regex	Regex	Local	DataEntry	Used to check whether user entered data is valid
TownRegex	Regex	Local	DataEntry	Used to check whether user entered data is valid
CountyRegex	Regex	Local	DataEntry	Used to check whether user entered data is valid
PostcodeRegex	Regex	Local	DataEntry	Used to check whether user entered data is valid
EmailRegex	Regex	Local	DataEntry	Used to check whether user entered data is valid
ContactRegex	Regex	Local	DataEntry	Used to check whether user entered data is valid
seat	String	Local	DataEntry & SeatingPlan	Used to send the name of the seat that is currently being booked to the DataEntry form
day	Integer	Local	DataEntry & SeatingPlan	Used to send the currently selected day to the DataEntry form
row	TableRow object	Local	DataEntry	Used to hold information before it is input into the Table
id	Integer	Local	DataEntry	Holds the value that was autoincremented when the data was added to the table
seatRow	TableRow object	Local	DataEntry	Used to update the correctTable and row of the SeatTables with the id number
username	String	Local	LoginForm	The entered text for the username is checked against this value

password	String	Local	LoginForm	The entered text for the password is checked against this value
mainForm	Form Object	Local	LoginForm	Used to display the MainForm form
aboutForm	Form Object	Local	MainForm	Used to display the AboutForm form
seatingPlan	Form Object	Local	MainForm	Used to display the SeatingPlan form
CRecords	Form Object	Local	MainForm	Used to display the ViewCRecords form
SRecords	Form Object	Local	MainForm	Used to display the ViewSRecords form
Form	String	Local	ReportForm	Used to tell which form was open before the ReportForm was opened
currentTab	String	Local	ReportForm	Used to tell which tab was selected on the ViewSRecords form before the ReportForm was opened
DisabledText	String	Local	SeatingPlan	Used so the text "Disabled" can be displayed in button tooltips
row	Table	Local	SeatingPlan	Contains the data of a row from one of the SeatTables so that it can be checked to see if the seat has been booked
dataEntry	Form Object	Local	SeatingPlan	Used to display the DataEntry form
t	ToolTip	Local	SeatingPlan	Used so that tooltips can be displayed when the mouse curser is over a button
s	String	Local	SeatingPlan	Contains the Name of the button that called the event
c	Class of object type	Local	SeatingPlan	Used to check all of the controls in the Tab (e.g button, textbox etc)
CustomerIDSearchRegex	Regex	Local	ViewCRecords & ViewSRecords	Holds the Regex that the search range is checked against.
field	String	Local	ViewCRecords & ViewSRecords	Used to know which field is selected to be filtered
filter	String	Local	ViewCRecords & ViewSRecords	Holds the filter that will be applied to the table
token	String	Local	ViewCRecords & ViewSRecords	Holds the text which has been entered
reportForm	Form Object	Local	ViewCRecords & ViewSRecords	Used to display the ReportForm form

All other variables are available in the .designer files. E.g. for the “SeatingPlan” form the designer file is “seating.designer.cs.”

Huw Talliss

Data Structures

CustomerTable

Field Name	Data Type	Size	Purpose
CustomerID	integer	4	Primary Key
Title	nvarchar	10	Title of the customer
FirstName	nvarchar	50	First Name of the customer
Surname	nvarchar	50	Surname of the customer
Date	datetime	8	Date and time when the order was placed
HouseNumName	nvarchar	50	The customer's house number or name
Address1	nvarchar	50	The address line 1 of the customer
Address2	nvarchar	70	The address line 2 of the customer
Town	nvarchar	50	The town of the customer's address
County	nvarchar	25	The county of the customer's address
Postcode	nvarchar	10	The postcode of the customer's address
Email	nvarchar	70	The customer's email address
ContactNum	nvarchar	12	The contact number of the customer

SeatTables

Field Name	Data Type	Size	Purpose
CustomerID	integer	4	The ID of the customer that has booked the seat
SeatID	nvarchar	100	The name of the seat, the primary key
Price	money	19	The price of the seat
Disabled	bit	1	Whether the seat has disabled access or not

Radio Buttons

Option	Purpose
CustomerID	Sorts data by CustomerID
SeatID	Sorts data by SeatID
Surname	Sorts data by Surname

Combo Boxes

For filtering

Option	Purpose
CustomerID	Changes the field which is filtered
SeatID	Changes the field which is filtered
Title	Changes the field which is filtered
First Name	Changes the field which is filtered
Surname	Changes the field which is filtered
HouseNumName	Changes the field which is filtered
Address1	Changes the field which is filtered
Address2	Changes the field which is filtered
Town	Changes the field which is filtered
County	Changes the field which is filtered
Postcode	Changes the field which is filtered
Email	Changes the field which is filtered
ContactNum	Changes the field which is filtered

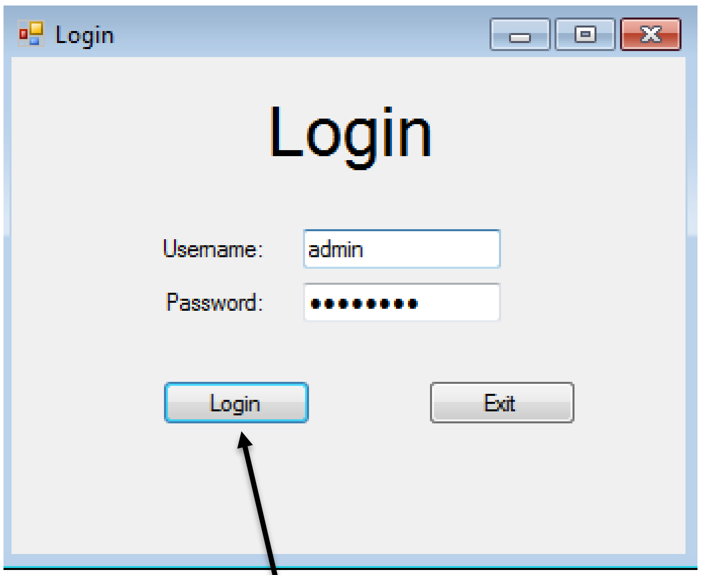
For data input

Option	Purpose
Mr.	Selects The Mr title
Miss.	Selects The Miss title
Mrs.	Selects The Mrs title
Ms.	Selects The Ms title
Dr.	Selects The Dr title
Prof.	Selects The Prof title

User Interface

Login Form

A simple username and password entry system prevents unauthorised access

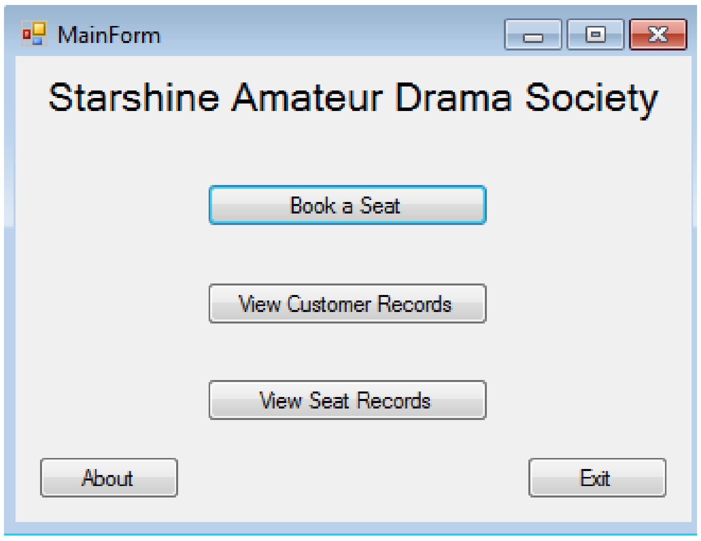


Username: admin
Password: ●●●●●●●●

The Button's name shows its function.

MainForm

This form allows easy navigation of the solution from one main location

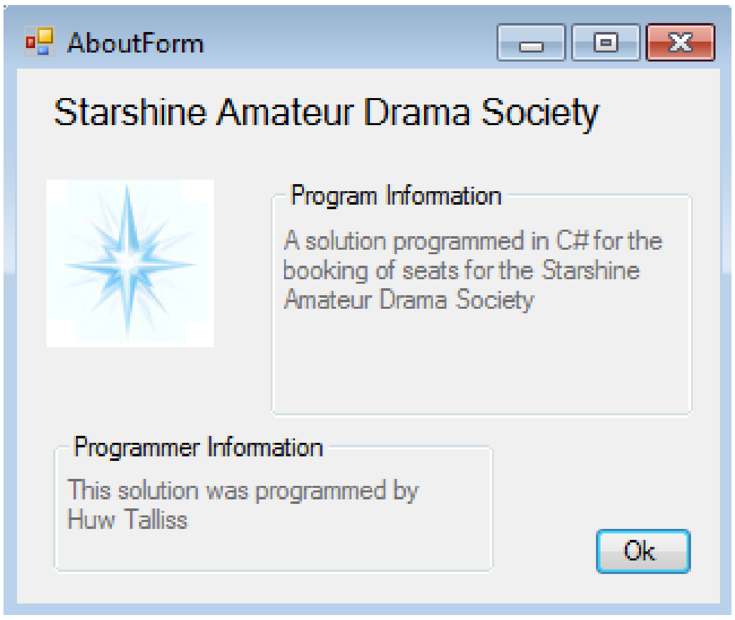


The buttons are displayed in an organised manner, making the program user friendly

Huw Talliss

AboutForm

Gives information about the solution as well as displaying the logo of SADS.


AboutForm

Starshine Amateur Drama Society

Program Information

A solution programmed in C# for the booking of seats for the Starshine Amateur Drama Society

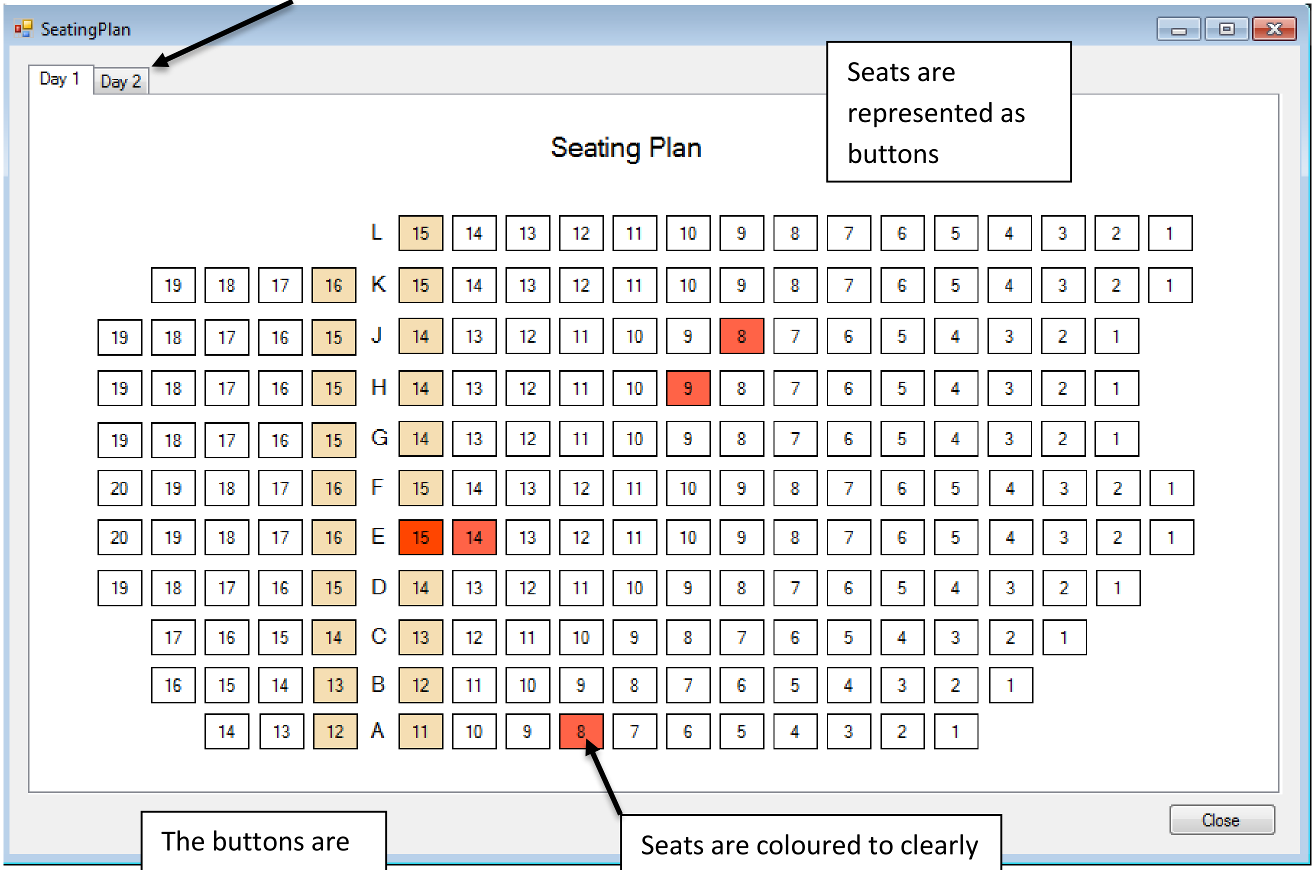
Programmer Information

This solution was programmed by Huw Talliss

SeatingPlan Form

Allows seats to be booked for different days

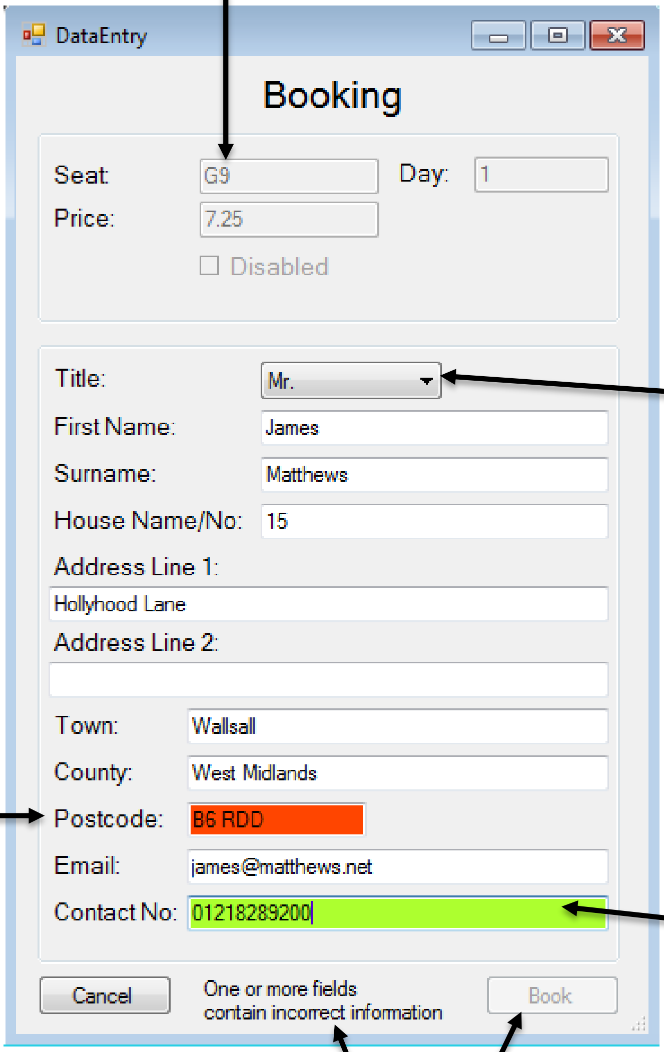
Seats are represented as buttons


SeatingPlan
Day 1
Day 2
Seating Plan
L 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
19 18 17 16 K 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
19 18 17 16 15 J 14 13 12 11 10 9 8 7 6 5 4 3 2 1
19 18 17 16 15 H 14 13 12 11 10 9 8 7 6 5 4 3 2 1
19 18 17 16 15 G 14 13 12 11 10 9 8 7 6 5 4 3 2 1
20 19 18 17 16 F 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
20 19 18 17 16 E 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
19 18 17 16 15 D 14 13 12 11 10 9 8 7 6 5 4 3 2 1
17 16 15 14 C 13 12 11 10 9 8 7 6 5 4 3 2 1
16 15 14 13 B 12 11 10 9 8 7 6 5 4 3 2 1
14 13 12 A 11 10 9 8 7 6 5 4 3 2 1
Close

The buttons are laid out like the seating plan

Seats are coloured to clearly show information about the seat. I.e. if they are booked or disabled

DataEntry Form



Booking

Seat:	G9	Day:	1
Price:	7.25		
	☐ Disabled		

Title:	Mr.
First Name:	James
Surname:	Matthews
House Name/No:	15
Address Line 1:	Hollyhood Lane
Address Line 2:	
Town:	Wallsall
County:	West Midlands
Postcode:	B6 RDD
Email:	james@matthews.net
Contact No:	01218289200

One or more fields contain incorrect information

The information of the seat is displayed here

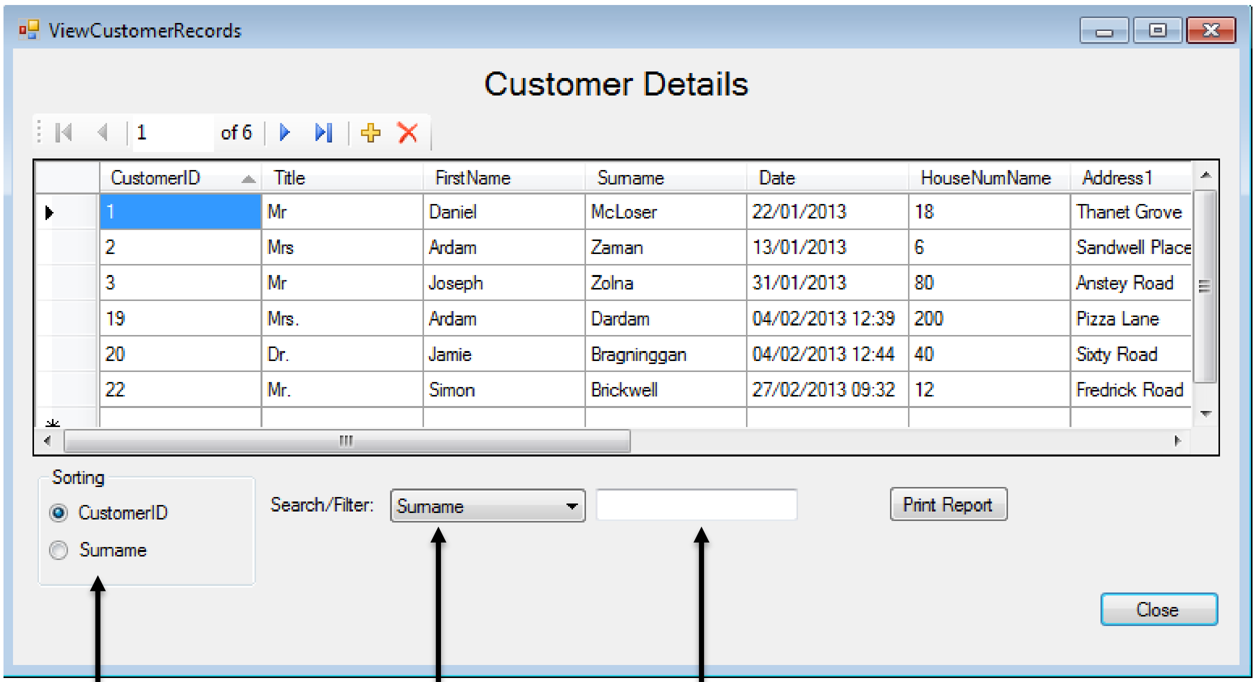
A dropdown menu limits the choice of the user

If a field's information is not allowed, turns red

If the field's information is allowed, it turns green until it is no longer selected

If any field's information is not allowed, the book button is disabled and an error message is shown at the bottom

ViewCRecords Form



ViewCustomerRecords

Customer Details

1 of 6

CustomerID	Title	FirstName	Surname	Date	HouseNumName	Address1
1	Mr	Daniel	McLoser	22/01/2013	18	Thanet Grove
2	Mrs	Ardam	Zaman	13/01/2013	6	Sandwell Place
3	Mr	Joseph	Zolna	31/01/2013	80	Anstey Road
19	Mrs.	Ardam	Dardam	04/02/2013 12:39	200	Pizza Lane
20	Dr.	Jamie	Bragninggan	04/02/2013 12:44	40	Sixty Road
22	Mr.	Simon	Brickwell	27/02/2013 09:32	12	Fredrick Road

Sorting
- [✓ CustomerID]
- Surname

Search/Filter: Surname

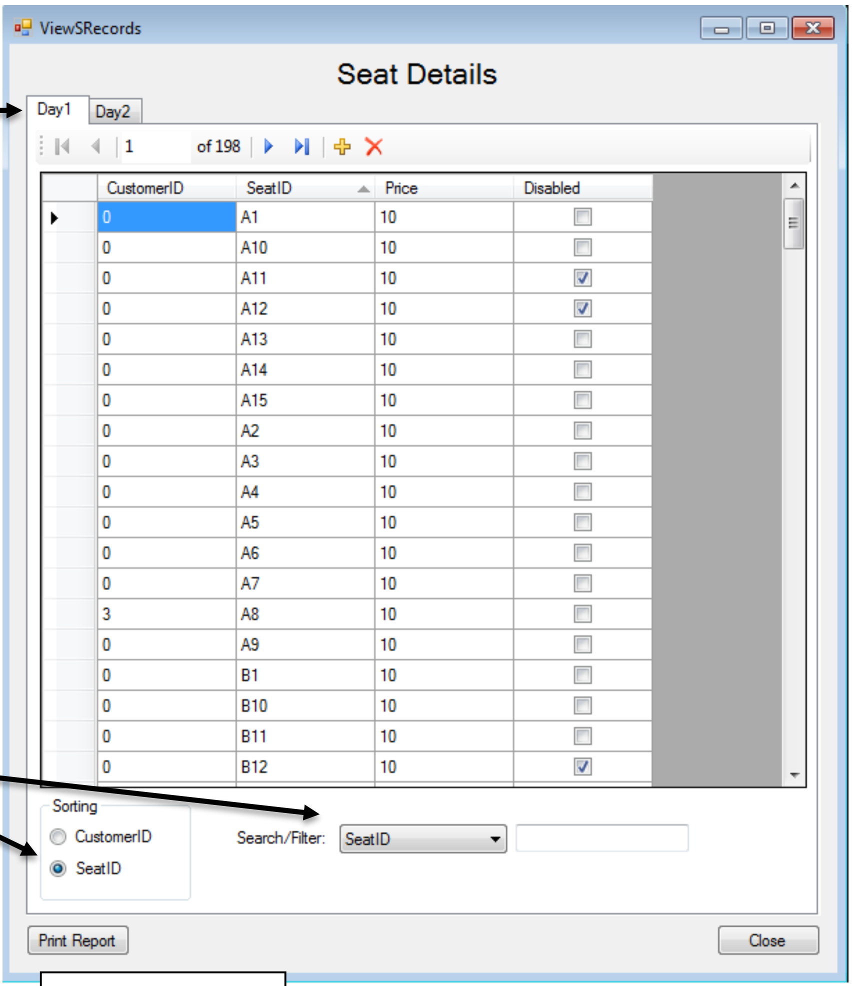
Radio Buttons used to sort the table

Dropdown menu to select which field to search by

Text box searches as you type

ViewSRecords Form

Using tabs lets multiple tables to be shown on the same form but still be simple to switch between



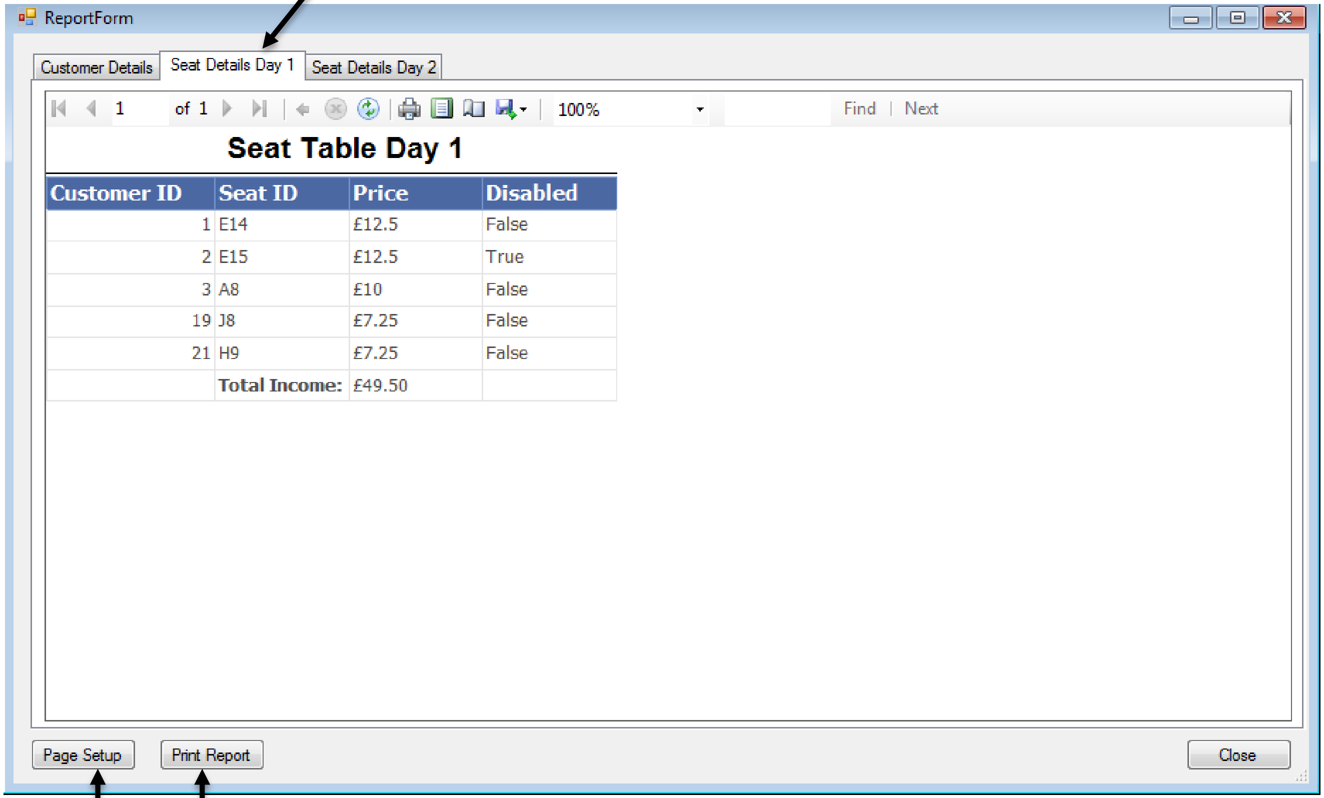
Seat Details

	CustomerID	SeatID	Price	Disabled
▶	0	A1	10	☐
	0	A10	10	☐
	0	A11	10	✓
	0	A12	10	✓
	0	A13	10	☐
	0	A14	10	☐
	0	A15	10	☐
	0	A2	10	☐
	0	A3	10	☐
	0	A4	10	☐
	0	A5	10	☐
	0	A6	10	☐
	0	A7	10	☐
	3	A8	10	☐
	0	A9	10	☐
	0	B1	10	☐
	0	B10	10	☐
	0	B11	10	☐
	0	B12	10	✓

Allows sorting and searching facilities

ReportForm

Tabs contain different reports for easy access



Seat Table Day 1

Customer ID	Seat ID	Price	Disabled
1	E14	£12.5	False
2	E15	£12.5	True
3	A8	£10	False
19	J8	£7.25	False
21	H9	£7.25	False
	Total Income:	£49.50	

Page setup and print buttons for easy printing

Annotated Listings

LoginForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace CG2_Solution
{
    public partial class LoginForm : Form
    {
        //The variables that the user input is checked agaisnt are declared here
        string username = "admin";
        string password = "password";

        public LoginForm()
        {
            InitializeComponent();
        }

        private void LoginForm_Load(object sender, EventArgs e)
        {

            UsernameTextBox.Text = "admin";
            PasswordTextBox.Text = "password";
        }

        private void ExitButton_Click(object sender, EventArgs e)
        {
            //Closes the form and application
            Close();
        }

        private void LoginButton_Click(object sender, EventArgs e)
        {
            //Checks the entered values agaisnt the constants username and password
            if (UsernameTextBox.Text == username && PasswordTextBox.Text == password)
            {
                //This shows the main form
                MainForm mainForm = new MainForm();
                mainForm.Show();
                //Creates an event
                mainForm.FormClosing += mainForm_FormClosing;
                //Hides this form
                this.Hide();
            }
            else
            {
                //Displays a message box if the entered values fail the test
                MessageBox.Show("Invalid Username or Password", "Error", MessageBoxButtons.OK, 
MessageBoxIcon.Error);
            }

        }

        void mainForm_FormClosing(object sender, FormClosingEventArgs e)
        {
            //Closes this form if the main form closes
            this.Close();
        }

        private void UsernameTextBox_TextChanged(object sender, EventArgs e)
        {

        }

        private void LoginForm_FormClosing(object sender, FormClosingEventArgs e)
        {

        }

        private void LoginForm_FormClosed(object sender, FormClosedEventArgs e)
        {

        }
    }
}
```

MainForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace CG2_Solution
{
    public partial class MainForm : Form
    {
```

Huw Talliss

```
    public MainForm()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {

    }

    private void MainForm_FormClosing(object sender, FormClosingEventArgs e)
    {

    }

    private void ExitButton_Click(object sender, EventArgs e)
    {
        //Closes the form
        this.Close();
    }

    private void AboutButton_Click(object sender, EventArgs e)
    {
        //Shows the about form
        AboutForm aboutForm = new AboutForm();
        aboutForm.Show();
        //Creates an event for when the aboutFormis closing
        aboutForm.FormClosing += aboutForm_FormClosing;
        //Hides this form
        this.Hide();
    }

    void aboutForm_FormClosing(object sender, FormClosingEventArgs e)
    {
        //When the aboutForm is closing, this form is shown
        this.Show();
    }



    private void ViewSeatsButton_Click(object sender, EventArgs e)
    {
        SeatingPlan seatingPlan = new SeatingPlan();
        seatingPlan.Show();
        seatingPlan.FormClosing += seatingPlan_FormClosing;
        this.Hide();
    }

    void seatingPlan_FormClosing(object sender, FormClosingEventArgs e)
    {
        this.Show();
    }

    private void CustomerRecsButton_Click(object sender, EventArgs e)
    {
        ViewCRecords CRecords = new ViewCRecords();
        CRecords.Show();
        CRecords.FormClosing += seatingPlan_FormClosing;
        this.Hide();
    }

    private void PrintReportButton_Click(object sender, EventArgs e)
    {
        //ReportForm reportView = new ReportForm();
        //reportView.Show();
        //reportView.FormClosing += reportView_FormClosing;
        //this.Hide();
    }

    void reportView_FormClosing(object sender, FormClosingEventArgs e)
    {
            this.Show();
        }

        private void SeatRecsButton_Click(object sender, EventArgs e)
        {
            ViewSRecords SRecords = new ViewSRecords();
            SRecords.Show();
            SRecords.FormClosing += SRecords_FormClosing;
            this.Hide();
        }

        void SRecords_FormClosing(object sender, FormClosingEventArgs e)
        {
            this.Show();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            //DataEntry dataEntry = new DataEntry();
            //dataEntry.Show();
            //dataEntry.FormClosing += dataEntry_FormClosing;
            //this.Hide();
        }

        void dataEntry_FormClosing(object sender, FormClosingEventArgs e)
        {
            this.Show();
        }

        private void BookSeatButton_Click(object sender, EventArgs e)
        {
            SeatingPlan seatingPlan = new SeatingPlan();
            seatingPlan.Show();
            seatingPlan.FormClosing += seatingPlan_FormClosing;
            this.Hide();
        }
    }

}
```

AboutForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace CG2_Solution
{
    public partial class AboutForm : Form
    {
        public AboutForm()
        {
            InitializeComponent();
        }

        private void CloseButton_Click(object sender, EventArgs e)
        {
            //Closes the form
            this.Close();
        }
    }
}
```

SeatingPlan.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace CG2_Solution
{
    public partial class SeatingPlan : Form
    {
        //Declares the string "DisabledText" and gives it a value
        string DisabledText = "Disabled";

        public SeatingPlan()
        {
            InitializeComponent();
        }

        private void CloseButton_Click(object sender, EventArgs e)
        {
            //Closes the form
            this.Close();
        }


        private void tabPage1_Click(object sender, EventArgs e)
        {

        }

        private void CloseButton_Click_1(object sender, EventArgs e)
        {
            this.Close();
        }

        private void SeatButton_Click(object sender, EventArgs e)
        {
            //If the selected tab is the "Day1Tab", the day variable is assigned the value 1 else, it is 
assigned the value 2
            int day = SeatingPlanTab.SelectedTab == Day1Tab ? 1 : 2;

            //The sender of the "SeatButton_Click" event's name is assigned to the "seat" variable
            string seat = ((Button)sender).Name;
            //Checks if the selected tab is "Day1Tab"
            if (SeatingPlanTab.SelectedTab == Day1Tab)
            {
                //Sets the variable "row" to the data of the seat that the button pressed corresponds 
with
                //This is so that it can be checked whether the seat is booked or not to prevent double 
booking
                var row = cG2DatabaseDataSet1.SeatTableDay1.FindBySeatID(((Button)sender).Name);
                //Checks if the selected seat's CustomerID field's value is not 0. If it is not 0, a
dialog is shown to say the
                //Seat is already booked and to allow the user to delete the booking.
                if (row.CustomerID != 0)
                {
                    //Displays dialog boxes to tell the user that the seat is booked and to let them
delete the booking
                    if (MessageBox.Show("This seat is currently booked. Would you like to delete this
booking?","Error", MessageBoxButtons.YesNo, MessageBoxIcon.Error) == DialogResult.Yes)
                    {
                        if (MessageBox.Show("Are you sure you want to delete this booking?", "Confirm
Selection", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
                        {
                            //If they clicked yes, the CustomerID of the row is set to 0 (Making it
unbooked)
                            row.CustomerID = 0;
                            //The table adapter is updated to apply the changes
                            seatTableDay1TableAdapter1.Update(row);
                            //If the seat's CustomerID is 0, a confirmation dialog is shown, this is to
check if the deletion succeded
                            if (row.CustomerID == 0)
                            {
                                MessageBox.Show("This booking has been deleted", "Confirmation message",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                            }
                            else
                            {
                                MessageBox.Show("An error has occurred and the booking has not been
deleted","Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                            }
                        }
                    }

                }
                else
                {
                    //If the seat's CustomerID is 0, the DataEntry form is shown.
                    //When it is called, it is passed the seat variable and the day variable so that the
data

                    //of the seat can be retrieved in that form
                    DataEntry dataEntry = new DataEntry(seat, day);
                    dataEntry.FormClosing += dataEntry_FormClosing;
                    dataEntry.ShowDialog();
                }
            }

            //If the Day1Tab is not selected, then the entire process occurs again but using the
SeatTableDay2 information instead
            else
            {
                //The first character of the name of the event's sender is removed before the row
variable is assigned
                var row = cG2DatabaseDataSet1.SeatTableDay2.FindBySeatID(((Button)sender).Name.Substring(
1));
                if (row.CustomerID != 0)
                {
                    if (MessageBox.Show("This seat is currently booked. Would you like to delete this
booking?","Error", MessageBoxButtons.YesNo, MessageBoxIcon.Error) == DialogResult.Yes)
                    {
                        if (MessageBox.Show("Are you sure you want to delete this booking?", "Confirm
Selection", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
                        {

                            row.CustomerID = 0;
                            seatTableDay2TableAdapter1.Update(row);

                            if (row.CustomerID == 0)
                            {
                                MessageBox.Show("This booking has been deleted", "Confirmation message",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                            }
                            else
                            {
                                MessageBox.Show("An error has occurred and the booking has not been
deleted","Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                            }
                        }
                    }
                }
                else
                {
                    DataEntry dataEntry = new DataEntry(seat, day);
                    dataEntry.ShowDialog();
                }
            }
        }

        void dataEntry_FormClosing(object sender, FormClosingEventArgs e)
        {
            //Runs the ApplyColors function to recheck the database in order to make sure all buttons are
the right colour
            ApplyColors();
        }


        private void SeatButton_MouseHover(object sender, EventArgs e)
        {
            //When you mouse over a seat button it displays a tooltip with the seat's information
            ToolTip t = new ToolTip();
            //Checks which tab is selected so that the correct information can be retrieved
            if (SeatingPlanTab.SelectedTab == Day1Tab)
            {
                try
                {
                    //Get the name of the button that called the event
                    string s = ((Button)sender).Name;
                    //Gets the information of the seat
                    var row = cG2DatabaseDataSet1.SeatTableDay1.FindBySeatID(s);
                    //Checks if the seat is a disabled seat so that it can be displayed in the tooltip
                    if (row.Disabled == false)
                    {
                        //Sets what the tooltip contains
                        t.SetToolTip((Button)sender, row.SeatID + " - £" + row.Price.ToString());
                    }
                    else
                    {
                        t.SetToolTip((Button)sender, row.SeatID + " - £" + row.Price.ToString() + " -
" +DisabledText);
                    }
                }
                catch
                {
                }
            }
            else
            {
                try
                {
                    //Gets the name of the button that called the event but removes the first character
                    string s = ((Button)sender).Name.Substring(1);
                    var row = cG2DatabaseDataSet1.SeatTableDay2.FindBySeatID(s);
                    if (row.Disabled == false)
                    {
                        t.SetToolTip((Button)sender, row.SeatID + " - £" + row.Price.ToString());
                    }
                    else
                    {
                        t.SetToolTip((Button)sender, row.SeatID + " - £" + row.Price.ToString() + " -
" +DisabledText);
                    }
                }
                catch
```

```
                {
                }
            }
        }

        private void SeatingPlan_Load(object sender, EventArgs e)
        {
            //Fills the table adapters so the information of the seats can be retrieved
            seatTableDay1TableAdapter1.Fill(cG2DatabaseDataSet1.SeatTableDay1);
            seatTableDay2TableAdapter1.Fill(cG2DatabaseDataSet1.SeatTableDay2);

            ApplyColors();

            //Testing adding values based on condition
            //var total = cG2DatabaseDataSet1.SeatTableDay1.Compute("Sum(Price)", "CustomerID<>0");
            //label25.Text = total.ToString();
        }


        private void ApplyColors()
        {
            //When the form is loaded, each object on the form is selected and if it is a button, then
the button's seat details
            //are checked and the colour of the button is changed appropriately
            foreach (Control c in Day1Tab.Controls)
            {
                if (c is Button)
                {
                    //Gets the information of the seat
                    var row = cG2DatabaseDataSet1.SeatTableDay1.FindBySeatID(((Button)c).Name);
                    //The colour of the button is changed dependent on whether the CustomerID is 0.If it
is 0 and the seat is a disabled seat
                    //the colour is changed to wheat. If it isnt disabled it's colour is set to white.
                    //If the CustomerID is not 0 and it is not disabled, the colour is set to tomato,
otherwise it is set to OrangeRed
                    c.BackColor = row.CustomerID == 0 ? (row.Disabled ? Color.Wheat : Color.White) : (row
.Disabled? Color.OrangeRed : Color.Tomato);
                }
            }
            foreach (Control c in Day2Tab.Controls)
            {
                if (c is Button)
                {
                    var row = cG2DatabaseDataSet1.SeatTableDay2.FindBySeatID(((Button)c).Name.Substring(1
));
                    c.BackColor = row.CustomerID == 0 ? (row.Disabled ? Color.Wheat : Color.White) : (row
.Disabled? Color.OrangeRed : Color.Tomato);
                }
            }
        }
    }
}
```

DataEntry.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Text.RegularExpressions;

namespace CG2_Solution
{
```

Huw Talliss

```
    public partial class DataEntry : Form
    {
        private string SeatID;
        private int Day;
        private float Price;
        private bool Disabled;

        //These are (mostly) all the same but are declared individually so they can be refactored easily
        private Regex FirstNameRegex = new Regex(@"^[a-zA-
ZāàáâäãåèéêëìíîïòóôöõøùúûüÿýñçčšžĀÀÁÂÄÃÅÈÉÊËÌÍÎÏÒÓÔÖÕØÙÚÛÜŸÝÑßÇŒÆČŠŽ∂ð ,.'-]+$");
        private Regex LastNameRegex = new Regex(@"^[a-zA-
ZāàáâäãåèéêëìíîïòóôöõøùúûüÿýñçčšžĀÀÁÂÄÃÅÈÉÊËÌÍÎÏÒÓÔÖÕØÙÚÛÜŸÝÑßÇŒÆČŠŽ∂ð ,.'-]+$");
        private Regex HouseNameNumRegex = new Regex(@"^[0-9a-zA-
ZāàáâäãåèéêëìíîïòóôöõøùúûüÿýñçčšžĀÀÁÂÄÃÅÈÉÊËÌÍÎÏÒÓÔÖÕØÙÚÛÜŸÝÑßÇŒÆČŠŽ∂ð ,.'-]+$");
        private Regex Address1Regex  = new Regex(@"^[0-9a-zA-
ZāàáâäãåèéêëìíîïòóôöõøùúûüÿýñçčšžĀÀÁÂÄÃÅÈÉÊËÌÍÎÏÒÓÔÖÕØÙÚÛÜŸÝÑßÇŒÆČŠŽ∂ð ,.'-]+$");
        private Regex Address2Regex = new Regex(@"^[0-9a-zA-
ZāàáâäãåèéêëìíîïòóôöõøùúûüÿýñçčšžĀÀÁÂÄÃÅÈÉÊËÌÍÎÏÒÓÔÖÕØÙÚÛÜŸÝÑßÇŒÆČŠŽ∂ð ,.'-]*$");
        private Regex TownRegex = new Regex(@"^[a-zA-
ZāàáâäãåèéêëìíîïòóôöõøùúûüÿýñçčšžĀÀÁÂÄÃÅÈÉÊËÌÍÎÏÒÓÔÖÕØÙÚÛÜŸÝÑßÇŒÆČŠŽ∂ð ,.'-]+$");
        private Regex CountyRegex = new Regex(@"^[a-zA-
ZāàáâäãåèéêëìíîïòóôöõøùúûüÿýñçčšžĀÀÁÂÄÃÅÈÉÊËÌÍÎÏÒÓÔÖÕØÙÚÛÜŸÝÑßÇŒÆČŠŽ∂ð ,.'-]+$");
        //The postcode Regex is the official Regex used by the UK Government
        private Regex PostcodeRegex = new Regex(@"(GIR 0AA)|((([A-Z-[QVX]][0-9][0-9]?)|(([A-Z-[QVX]][A-Z-
[IJZ]][0-9][0-9]?)|(([A-Z-[QVX]][0-9][A-HJKSTUW])|([A-Z-[QVX]][A-Z-[IJZ]][0-9][ABEHMNPRVWXY]))))\s?[0-
9][A-Z-[CIKMOV]]{2})");
        private Regex EmailRegex = new Regex(@"^[A-Za-z0-9._%-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}$");
        private Regex
ContactRegex = new Regex(@"^((\(?0\d{4}\)?\s?\d{3}\s?\d{3})|(\(?0\d{3}\)?\s?\d{3}\s?\d{4})|(\(?0\d{2}\)?\
s?\d{4}\s?\d{4}))(\s?\#(\d{4}|\d{3}))?$");



        public DataEntry(string seat, int day)
        {
            InitializeComponent();

            //If the day is day 2 then the first character of the seat string is removed
            if (day == 2)
            {
                SeatID = seat.Substring(1);
            }
            else
            {
                SeatID = seat;
            }
            //Assigns the "day" value from the previous for to the "Day" variable so it can be used in
other functions of this form
            Day = day;

        }

        private void CloseBtn_Click(object sender, EventArgs e)
        {
            //Closes the form
            this.Close();
        }

        private void DataEntry_Load(object sender, EventArgs e)
        {
            //Fills the table adapters with the tables
            customerTableTableAdapter1.Fill(cG2DatabaseDataSet1.CustomerTable);
            seatTableDay1TableAdapter1.Fill(cG2DatabaseDataSet1.SeatTableDay1);
            seatTableDay2TableAdapter1.Fill(cG2DatabaseDataSet1.SeatTableDay2);

            //enables the book button
            BookBtnEnable();
            //Sets the text of the ValidationErrorLabel to nothing
            ValidationErrorLabel.Text = "";
            //Sets the text of these text boxes to the seat that is being booked and for which day is
being booked
            SeatTBox.Text = SeatID;
            DayTBox.Text = Day.ToString();

            /*Dependant on which day is being booked, the seat table for that day is checked to get the
price and the disabled
            status of the the seat for that day*/
            if (Day == 1)
            {
                //try
                {
                    Price = (float)cG2DatabaseDataSet1.SeatTableDay1.FindBySeatID(SeatID).Price;
                    Disabled = cG2DatabaseDataSet1.SeatTableDay1.FindBySeatID(SeatID).Disabled;
                }
            }
            else
            {
                //try
                {
                    Price = (float)cG2DatabaseDataSet1.SeatTableDay2.FindBySeatID(SeatID).Price;
                    Disabled = cG2DatabaseDataSet1.SeatTableDay2.FindBySeatID(SeatID).Disabled;
                }
            }
            //Sets the text of the price textbox and whether the disabled checkbox is ticked by the
values just retrieved from the table
            PriceTBox.Text = Price.ToString();
            disabledCheckBox.Checked = Disabled;
        }

        private void BookBtn_Click(object sender, EventArgs e)
        {
            //When the Book Button is clicked, the information the user has entered is added into a new
row in the customer table
            CG2DatabaseDataSet.CustomerTableRow row = cG2DatabaseDataSet1.CustomerTable.NewCustomerTableR
ow();
            //As the TitleBox is a combobox, the selected item must be converted into a string
            row.Title = TitleBox.SelectedItem.ToString();
            row.FirstName = FirstnameBox.Text;
            row.Surname = SurnameBox.Text;
            //Sets row.Date to the current time
            row.Date = DateTime.Now;
            row.HouseNumName = HouseBox.Text;
            row.Address1 = Address1Box.Text;
            row.Address2 = Address2Box.Text;
            row.Town = TownBox.Text;
            row.County = CountyBox.Text;
            row.Postcode = PostcodeBox.Text;
            row.Email = EmailBox.Text;
            row.ContactNum = ContactBox.Text;

            //Declares an integer that will be used to contain the autoincremented ID of the new row
            int id;

            /*Normally, if you tried to add the new row and then get the incremented ID of the row, you
would not be able to
            as a connection to the table adapter would be opened, the row would be added, then it would
be closed before the
            command to get the ID would happen, meaning you could not get the ID.

            So, the connection is first opened, then the row is inserted. Then the ID of the new row is
retrieved and assigned
            to the "id" variable. Then the connection is closed.
            */
            try
            {
                //If the conection is not open, open a connection
                if (customerTableTableAdapter1.Connection.State != ConnectionState.Open)
                    customerTableTableAdapter1.Connection.Open();

                //Inserts a new row
                customerTableTableAdapter1.Insert(row.Title, row.FirstName, row.Surname, row.Date,
row.HouseNumName, row.Address1, row.Address2, row.Town, row.County, row.Postcode, row.Email,
row.ContactNum);
                //Retrieves the value that was autoincremented when the row was inserted and assigns it
to "id"
                id = Convert.ToInt32(customerTableTableAdapter1.GetIdentity());
            }
            finally
            {
                //If the conection is not closed, close the connection
                if (customerTableTableAdapter1.Connection.State != ConnectionState.Closed)
                    customerTableTableAdapter1.Connection.Close();
            }

            // MessageBox.Show("Added row with id = " + id.ToString());

            //The id that was just retrieved is added to the CustomerID of the seat that has been booked
            if (Day == 1)
            {
                //Finds the row of the seat that is being booked by using the SeatID of it
                CG2DatabaseDataSet.SeatTableDay1Row seatRow =cG2DatabaseDataSet1.SeatTableDay1.FindBySeat
ID(SeatID);
                seatRow.CustomerID = id;
                //The table adapter is updated, to apply the changes
                seatTableDay1TableAdapter1.Update(seatRow);
            }
            else
            {
                CG2DatabaseDataSet.SeatTableDay2Row seatRow =cG2DatabaseDataSet1.SeatTableDay2.FindBySeat
ID(SeatID);
                seatRow.CustomerID = id;
                seatTableDay2TableAdapter1.Update(seatRow);
            }

            if (MessageBox.Show("The seat has been successfully booked", "Confirmation message",
MessageBoxButtons.OK, MessageBoxIcon.Information) == DialogResult.OK)
            {
                this.Close();
            }
        }

        //The Enable book button function. This validates the information entered and will only enable
the Book Button if all fields vallidate correctly
        //It is called whenever the text of a textbox changes
        private void BookBtnEnable()
        {
            //Matches each field to its Regular expression regex
            if (FirstNameRegex.IsMatch(FirstnameBox.Text) && LastNameRegex.IsMatch(SurnameBox.Text) &&Hou
seNameNumRegex.IsMatch(HouseBox.Text) && Address1Regex.IsMatch(Address1Box.Text) &&Address2Regex.IsMatch(
Address2Box.Text) && TownRegex.IsMatch(TownBox.Text) && CountyRegex.IsMatch(CountyBox.Text)&& PostcodeReg
ex.IsMatch(PostcodeBox.Text) && EmailRegex.IsMatch(EmailBox.Text) &&ContactRegex.IsMatch(ContactBox.Text)
 == true)
            {
                //Enables the Book Button
                BookBtn.Enabled = true;
                ValidationErrorLabel.Text = "";
            }
            else
            {
                //Disables the book button
                BookBtn.Enabled = false;
                ValidationErrorLabel.Text = "One or more fields \ncontain incorrect information";
            }
        }


        private void ValidatableTextBox_Leave(object sender, EventArgs e)
        {
            //Checks if the textbox that lost focus is white or green or empty and, if it is, its colour
is set to white. If it is not, the text box's colour is set to OrangeRed
```

```
            ((TextBox)sender).BackColor = (((TextBox)sender).BackColor == Color.White ||((TextBox)sender)
.BackColor == Color.GreenYellow || ((TextBox)sender).Text.Length == 0) ? Color.White :Color.OrangeRed;
            //The BookBtnEnable function is called
            BookBtnEnable();
        }

        private void FirstnameBox_TextChanged(object sender, EventArgs e)
        {
            //If the Regular Expression Regex matches the text entered then the textbox that called the
event's colour is changed to GreenYellow. If it does not match, the colour is changed to OrangeRed
            ((TextBox)sender).BackColor = FirstNameRegex.IsMatch(((TextBox)sender).Text) ? Color.GreenYel
low :Color.OrangeRed;
            BookBtnEnable();
        }

        private void SurnameBox_TextChanged(object sender, EventArgs e)
        {
            ((TextBox)sender).BackColor = LastNameRegex.IsMatch(((TextBox)sender).Text) ? Color.GreenYell
ow :Color.OrangeRed;
            BookBtnEnable();
        }

        private void HouseBox_TextChanged(object sender, EventArgs e)
        {
            ((TextBox)sender).BackColor = HouseNameNumRegex.IsMatch(((TextBox)sender).Text) ? Color.Green
Yellow :Color.OrangeRed;
            BookBtnEnable();
        }

        private void Address1Box_TextChanged(object sender, EventArgs e)
        {
            ((TextBox)sender).BackColor = Address1Regex.IsMatch(((TextBox)sender).Text) ? Color.GreenYell
ow :Color.OrangeRed;
            BookBtnEnable();
        }

        private void Address2Box_TextChanged(object sender, EventArgs e)
        {
            ((TextBox)sender).BackColor = Address2Regex.IsMatch(((TextBox)sender).Text) ? Color.GreenYell
ow :Color.OrangeRed;
            BookBtnEnable();
        }

        private void TownBox_TextChanged(object sender, EventArgs e)
        {
            ((TextBox)sender).BackColor = TownRegex.IsMatch(((TextBox)sender).Text) ? Color.GreenYellow :
Color.OrangeRed;
            BookBtnEnable();
        }

        private void CountyBox_TextChanged(object sender, EventArgs e)
        {
            ((TextBox)sender).BackColor = CountyRegex.IsMatch(((TextBox)sender).Text) ? Color.GreenYellow
 :Color.OrangeRed;
            BookBtnEnable();
        }

        private void PostcodeBox_TextChanged(object sender, EventArgs e)
        {
            ((TextBox)sender).BackColor = PostcodeRegex.IsMatch(((TextBox)sender).Text) ? Color.GreenYell
ow :Color.OrangeRed;
            BookBtnEnable();
        }

        private void EmailBox_TextChanged(object sender, EventArgs e)
        {
            ((TextBox)sender).BackColor = EmailRegex.IsMatch(((TextBox)sender).Text) ? Color.GreenYellow
:Color.OrangeRed;
            BookBtnEnable();
        }
        private void ContactBox_TextChanged(object sender, EventArgs e)
        {
            ((TextBox)sender).BackColor = ContactRegex.IsMatch(((TextBox)sender).Text) ? Color.GreenYello
w :Color.OrangeRed;
            BookBtnEnable();

        }
    }
}
```

ViewCRecords.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Text.RegularExpressions;

namespace CG2_Solution
{
    public partial class ViewCRecords : Form
    {
        public ViewCRecords()
        {
            InitializeComponent();
            //Sets the SearchComboBox's currently selected item to the surname item as this is often the
field people will
            //want to search by
            SearchComboBox.SelectedItem = "Surname";
        }
        //Declares a Regular Expression Regex to be used for the filtering/searching
        private Regex CustomerIDSearchRegex = new Regex(@"^[0-9]+$");

        private void ViewCRecords_Load(object sender, EventArgs e)
        {
            //This line of code loads data into the 'cG2DatabaseDataSet1.CustomerTable' table.
            customerTableTableAdapter.Fill(this.cG2DatabaseDataSet1.CustomerTable);
            //Checks the CustomerID radio button
            CustomerIDSortBtn.Checked = true;
        }

        private void PrintReportBtn_Click(object sender, EventArgs e)
        {
            //Shows the report view form and hides the current form
            //ReportView reportView = new ReportView();
            //reportView.Show();
            //reportView.FormClosing += reportView_FormClosing;
            //this.Hide();
        }

        void reportView_FormClosing(object sender, FormClosingEventArgs e)
        {
            //Shows this form when the report view form is closing
            this.Show();
        }

        private void CloseBtn_Click(object sender, EventArgs e)
        {
            //Saves the changes to the database
            this.customerTableTableAdapter.Update(this.cG2DatabaseDataSet1);
            this.Close();
        }


        private void SurnameSortBtn_CheckedChanged(object sender, EventArgs e)
        {
            //When the SurnameSort button's check value is changed, the table is sorted by the surname
field
            CustomerTableDataGridView.Sort(surnameDataGridViewTextBoxColumn,
ListSortDirection.Ascending);
        }

        private void CustomerIDSortBtn_CheckedChanged(object sender, EventArgs e)
        {
            //When the CustomerID button's check value is changed, the table is sorted by the CustomerID
field
            CustomerTableDataGridView.Sort(customerIDDataGridViewTextBoxColumn,
ListSortDirection.Ascending);
        }

        private void SearchBox_TextChanged(object sender, EventArgs e)
        {
            //When the text in the SearchBox changes, the search filter is applied to the table if the
entered text is valid
            //Creating variables to be used to contain the field that the filter will be applied to, the
filter that should be used
            //and the entered text
            string field = SearchComboBox.SelectedItem.ToString();
            string filter = "";
            string token = ((TextBox)sender).Text;

            //As the CustomerID field is an integer type, a seperate filter needs to be used otherwise it
would throw an error
            //so the currently selected field is determined and if it is the CustomerID field, the regex
earlier defined is used
            //so that the filter will only be applied if the entered text contains numbers only
            if (field == "CustomerID")
            {
                //Checks the entered text against the Regular Expression
                if (CustomerIDSearchRegex.IsMatch(token))
                {
                    //Declares the filter
                    filter = token == "" ? "" : "(" + field + " = " + token + ")";
                }
            }
            else
            {
                filter = token == "" ? "" : "(" + field + " LIKE '%" + token + "%')";
            }
            //Applies the filter to the table
            customerTableBindingSource.Filter = filter;
        }

        private void PrintReportBtn_Click_1(object sender, EventArgs e)
        {
            //Shows the report form while keeping this form open, sending the reportForm the string
"CustomerTable"
            ReportForm reportForm = new ReportForm("CustomerTable");
            reportForm.ShowDialog();

            //ReportView reportView = new ReportView();
            //reportView.ShowDialog();
            //reportView.FormClosing += reportView_FormClosing;
        }
    }
}
```

ViewSRecords.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Text.RegularExpressions;

namespace CG2_Solution
{
    public partial class ViewSRecords : Form
    {
        public ViewSRecords()
        {
            InitializeComponent();
        }

        //Declares a Regular Expression Regex to be used for the filtering/searching
        private Regex CustomerIDSearchRegex = new Regex(@"^[0-9\.]+$");

        private void ViewSRecords_Load(object sender, EventArgs e)
        {
            //Loads data into the table
            seatTableDay1TableAdapter1.Fill(this.cG2DatabaseDataSet1.SeatTableDay1);
            seatTableDay2TableAdapter1.Fill(this.cG2DatabaseDataSet1.SeatTableDay2);

            //Sorts the tables by the SeatID field
            SeatTable1DatagridView.Sort(seatIDDataGridViewTextBoxColumn, ListSortDirection.Ascending);
            SeatTable2DatagridView.Sort(seatIDDataGridViewTextBoxColumn1, ListSortDirection.Ascending);

            //Selects SeatID in both combo boxes
            SearchComboBox1.SelectedItem = "SeatID";
            SearchComboBox2.SelectedItem = "SeatID";
        }

        private void CloseBtn_Click(object sender, EventArgs e)
        {
            //Closes the form
            this.Close();
        }

        //Sorting for Tab 1
        private void CustomerIDSortBtn_CheckedChanged(object sender, EventArgs e)
        {
            //When the button is checked, the table is sorted by the CustomerID field
            SeatTable1DatagridView.Sort(customerIDDataGridViewTextBoxColumn,
ListSortDirection.Ascending);
        }

        private void SeatIDSortBtn_CheckedChanged(object sender, EventArgs e)
        {
            //When the button is checked, the table is sorted by the SeatID field
            SeatTable1DatagridView.Sort(seatIDDataGridViewTextBoxColumn, ListSortDirection.Ascending);
        }

        //Sorting for Tab 2
        private void CustomerIDSortBtn2_CheckedChanged(object sender, EventArgs e)
        {
            SeatTable2DatagridView.Sort(customerIDDataGridViewTextBoxColumn1,
ListSortDirection.Ascending);
        }

        private void SeatIDSortBtn2_CheckedChanged(object sender, EventArgs e)
        {
            SeatTable2DatagridView.Sort(seatIDDataGridViewTextBoxColumn1, ListSortDirection.Ascending);
        }

        private void SearchBox1_TextChanged(object sender, EventArgs e)
        {
            //When the text in the SearchBox changes, the search filter is applied to the table if the
entered text is valid
            //Creating variables to be used to contain the field that the filter will be applied to, the
filter that should be used
            //and the entered text
            string field = SearchComboBox1.SelectedItem.ToString();
            string filter = "";
            string token = ((TextBox)sender).Text;

            //As the CustomerID field is an integer type, a seperate filter needs to be used otherwise it 
would throw an error
            //so the currently selected field is determined and if it is the CustomerID field, the regex 
earlier defined is used
            //so that the filter will only be applied if the entered text contains numbers only
            if (field == "CustomerID" || field == "Price")
            {
                //Checks the entered text against the Regular Expression
                if (CustomerIDSearchRegex.IsMatch(token))
                {
                    //Declares the filter
                    filter = token == "" ? "" : "(" + field + " = " + token + ")";
                }
            }
            else
            {
                filter = token == "" ? "" : "(" + field + " LIKE '%" + token + "%')";
            }
            //Applies the filter to the table
            seatTableDay1BindingSource.Filter = filter;
        }

        private void SearchBox2_TextChanged(object sender, EventArgs e)
        {
            string field = SearchComboBox2.SelectedItem.ToString();
            string filter = "";
            string token = ((TextBox)sender).Text;
            if (field == "CustomerID" || field == "Price")
            {
                if (CustomerIDSearchRegex.IsMatch(token))
                {
                    filter = token == "" ? "" : "(" + field + " = " + token + ")";
                }
            }
            else
            {
                filter = token == "" ? "" : "(" + field + " LIKE '%" + token + "%')";
            }
            seatTableDay2BindingSource.Filter = filter;
        }

        private void PrintReportBtn_Click(object sender, EventArgs e)
        {
            //Checks which tab is currently selected so the correct string can be passed to the 
reportform when
            if (SeatRecordsTab.SelectedTab == Day1Tab)
            {
                //Shows the reportForm passing it the string "SeatTableDay1"
                ReportForm reportForm = new ReportForm("SeatTableDay1");
                reportForm.ShowDialog();
            }

            if (SeatRecordsTab.SelectedTab == Day2Tab)
            {
                ReportForm reportForm = new ReportForm("SeatTableDay2");
                reportForm.ShowDialog();
            }
        }
    }
}
```

ReportForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace CG2_Solution
{
    public partial class ReportForm : Form
    {
        //Declares variables that are used in deciding which tab is open
        string Form;
        string currentTab;

        public ReportForm(string whichForm)
        {
            //Assigns the value passed to the form when it is initialised to the value Form
            Form = whichForm;
            InitializeComponent();
        }

        private void ReportForm_Load(object sender, EventArgs e)
        {
            //This line of code loads data into the 'CG2DatabaseDataSet.SeatTableDay2' table.
            this.SeatTableDay2TableAdapter.Fill(this.CG2DatabaseDataSet.SeatTableDay2);
            //This line of code loads data into the 'CG2DatabaseDataSet.CustomerTable' table.
            this.CustomerTableTableAdapter.Fill(this.CG2DatabaseDataSet.CustomerTable);
            //This line of code loads data into the 'CG2DatabaseDataSet.SeatTableDay1' table.
            this.SeatTableDay1TableAdapter.Fill(this.CG2DatabaseDataSet.SeatTableDay1);

            //Refreshes all of the reportviewers in the form
            this.CustomerReportViewer.RefreshReport();
            this.SeatDay1ReportViewer.RefreshReport();
            this.SeatDay2ReportViewer.RefreshReport();
            //this.CustomerReportViewer.PrinterSettings

            //Checks the value of "Form" to decide which tab will be selected when the form first opens
            if (Form == "CustomerTable")
            {
                //Selects the 1st tab. The tab index starts at 0
                ReportTabControl.SelectedIndex = 0;
            }
            if (Form == "SeatTableDay1")
            {
                ReportTabControl.SelectedIndex = 1;
            }
            if (Form == "SeatTableDay2")
            {
                ReportTabControl.SelectedIndex = 2;
            }


        }

        private void CloseBtn_Click(object sender, EventArgs e)
        {
            //Closes the form
            this.Close();
        }

        private void printReportBtn_Click(object sender, EventArgs e)
        {
            //Gets the name of the currently selected tab and assigns "currentTab" to it
            currentTab = ReportTabControl.SelectedTab.Name;
            //Checks the value of "currentTab"
            if (currentTab == "CustomerTab")
            {
                //Brings up the print dialog for that specific report
                CustomerReportViewer.PrintDialog();
            }
            if (currentTab == "SeatDay1Tab")
            {
                SeatDay1ReportViewer.PrintDialog();
            }
            if (currentTab == "SeatDay2Tab")
            {
                SeatDay2ReportViewer.PrintDialog();
            }
        }

        private void PagesetupBtn_Click(object sender, EventArgs e)
        {
            //Gets the name of the currently selected tab and assigns "currentTab" to it
            currentTab = ReportTabControl.SelectedTab.Name;
            //Checks the value of "currentTab"
            if (currentTab == "CustomerTab")
            {
                //Brings up the page setup dialog for that specific report
                CustomerReportViewer.PageSetupDialog();
            }
            if (currentTab == "SeatDay1Tab")
            {
                SeatDay1ReportViewer.PageSetupDialog();
            }
            if (currentTab == "SeatDay2Tab")
            {
                SeatDay2ReportViewer.PageSetupDialog();
            }
        }
    }
}
```