

## **Problem Definition**

### **Background information**

The Starshine amateur dramatic society has grown in size recently and has decided to introduce a third show into their annual schedule of shows. The third show will be performed in early Autumn, and their normal venue- a church hall- is unavailable for the performance so they are going to use a local theatre. As the theatre is very big, and in order to cover the costs of booking it, they will need to sell a large amount of tickets. The dates of the performances are the Friday and Saturday of the first week of October.

The society would like a system that allows them to easily process booking orders and work out their total profits.

### **Main Aims**

The program I create will do the following:

- Allow customer details such as name, email, address etc. to be input and stored into a database.
- Display all the available seats and show if they have disabled access or not so that it is easy to tell which seats are available.
- Enable seats to be booked for each day so that the same seat can have multiple days on which it is booked.
- Ensure that seats are not double booked as this would cause several problems.
- Store and retrieve the details of all the bookings made by a customer for use in contacting the customer and viewing total income.
- Output the total ticket sales for each performance so the company can tell if they will make a profit.

### **Limitations of the program**

- The program will not be able to be used across a network as data would be split up and it would not be possible to ensure seats are not double booked.
- The databases will not be secure against non-authorized users. It is up to the company to ensure their computers are secured.
- The program will not be usable on non-Windows operating systems or Windows systems running on Windows 95 or earlier.
- I will not be able to keep the system free from errors made by the human typist that follow the validation rules I will have. The user will have to check for mistakes in entered information.

### **Assumptions made**

- The seats with disabled access may also be used by able-bodied people.
- There are no group discounts/disabled discounts.
- Tickets will be paid for at the performance itself.

## Objectives

The system I am going to create will:

- Data input
  - Have a form which allows data to be added to and deleted from the table.
  - Have a save button which saves the record to the table.
  - Have a button to cancel data entry.
  - Include validation for data like post codes and emails to ensure the entered data is valid.
- Data Manipulation
  - A database will be created to store several tables that hold customer records, booking information etc.
  - Allow the user to search for a specific record.
  - Allow the user to sort the table based on a specified field.
  - The system will be able to calculate the total income for each performance.
- Interface
  - Provide a simple and user friendly interface.
  - Have a visual display of the seating plan and if seats are booked or not.
  - Clicking on an un-booked seat directly leads to an order form.
  - The hint feature will be used to tell users what different buttons do.
  - There will be labels that simply tell users what things are.
  - There will be meaningful field labels.
  - Suitable error messages will be included to make it clear to a user what the problem is in simple language so that they may understand it or be able to solve it.
  - I will avoid the colours red and green so that the system can still be used easily by those who are colour blind.
- Output
  - Allow several different reports to be created and displayed on the monitor.
  - Allow these reports to be printed off.
- Security
  - Upon loading the program, there will be a login form before access to the system is granted.

## Justification

I will be using the high level language C# with the .NET framework for my solution because:

- It has a powerful GUI facility which will allow me to create a professional looking program and if needed, implement pictures to further improve the aesthetics of the program.
- C# is a high level language and is easier to program with than low level languages such as assembly. It is also object orientated, further making it easier to program in as visual objects are used as well as the actual code.
- C# also has a large number of libraries which allow many different calculations to be used simply. It is also an industry standard language.
- I will also be using the .NET framework which is a very strong framework, provided by Microsoft. It has many controls that can be easily implemented and also allows easy connection to a database.
- For reports, I will be using Microsoft ReportViewer. It is also provided by Microsoft, it is a very stable and easy to implement with the .NET framework.
- My database will be an SQL database. SQL is an ANSI Standard, widely used and very stable. Many other programs can edit SQL databases meaning the database can be used elsewhere if needed.

**Data structures and methods of access**

My solution will use two sets of two relational tables, one set for each performance. One will store customer details, one will store seat details.

**Customer Details Table**

This table will store the information of the customer for each different seat that is booked. The table will grow up to a size of 197 records as that is the number of seats available.

Field Name	Data type	Description	Length	Example
(Primary Key) Customer ID	integer	An auto incrementing integer	0	21
(Foreign Key) Seat ID	String	The seat's ID	3	E4
Title	String	The title of the customer	4	Mr
First Name	String	The customer's first name	25	Daniel
Surname	String	The customers surname	30	James
Date	Date	The date of the order	8	17/10/12
House Number/Name	String	The house name or number of the customer's address	30	10
Address 1	String	The first address line of the customer's address	70	Thanet Grove
Address 2	String	The second address line of the customer's address, if they live in an apartment building	70	Apartment B3
Town	String	The town of the customer's address	50	Perry Barr
County	String	The county of the customer's address	25	West Midlands
Post Code	String	The Post Code of the customer	10	B42 1SS
Email	String	The customer's email	70	me@example.com
Contact Number	String	The contact number of the customer	9	01213621007

### Seat Details Table

This table stores the information for each seat, such as the price of the seat and whether it has disabled access or not. This table will permanently have the same number of records- the number of seats, (197). This way of handling the seats allows for the system to be used with different seating plans or more seats. It also allows for the price and other data to be edited for each individual seat.

Field Name	Data type	Description	Length	Example
(Foreign Key) Customer ID	integer	An auto incrementing integer	0	21
(Primary Key) Seat ID	String	The Seat's ID	3	E4
Price	Currency	The price of the seat	4	£12.50
Disabled	Boolean	Whether the seat has disabled access	1	1

### Sorting

Sorting will only be implemented for certain key fields, the Customer ID, Seat ID and Surname for the Customer Details Table, and the Customer ID and Seat ID for the Seat Details Table. This will avoid overcrowding the sort function by only allowing sorting of important fields.

### Searching

Similarly to sorting, searching will only be provided for certain important fields. Customer ID and Surname for the Customer Details Table, and the Customer ID and Seat ID for the Seat Details Table. The search method will be linear as the database will not become a very large size.

### Validation

There are several features build into the .NET framework that already include validation:

- Check Boxes – These can either be checked or be left blank, being the ideal item for fields with Boolean values.
- Combo Box – These can have a set of values stored in an array that can be selected by the use of a drop down list. These are useful for questions with categorical variables.

In addition to the build in validation, I will also include coded validation for text boxes using regular expressions. So, for example, only numbers may be input. If characters were input, the system would show an error. This would increase the user friendliness. Also, in fields such as First names and surnames, the first letter can be automatically changed to uppercase as to avoid excess error messages which may confuse the user. Other validation I will use is such as checking that fields are not left empty or that data entered is greater than zero for a certain field.

### Method of access

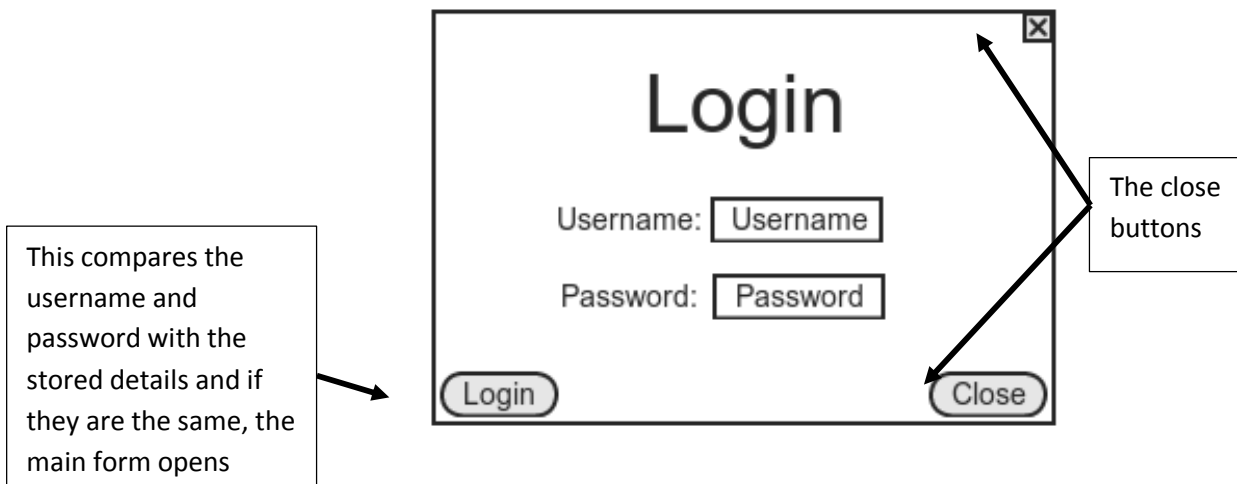
The three main databases for Windows are Access (Microsoft), Paradox (Borland) and Dbase (dBase Inc). They are all relational databases, where tables (files) can be linked by key fields.

Between the program and a database table, stands a Database Management System (DMBS), this is called Microsoft SQL Server. It handles all requests to read from or write to its table(s). It will stop you doing things you shouldn't by generating Database Engine errors. On the plus side, it takes care of your data. You don't need to know about how saving or loading takes place. It provides you with search and query functions. It allows you to store pictures as well as numerical or string data.

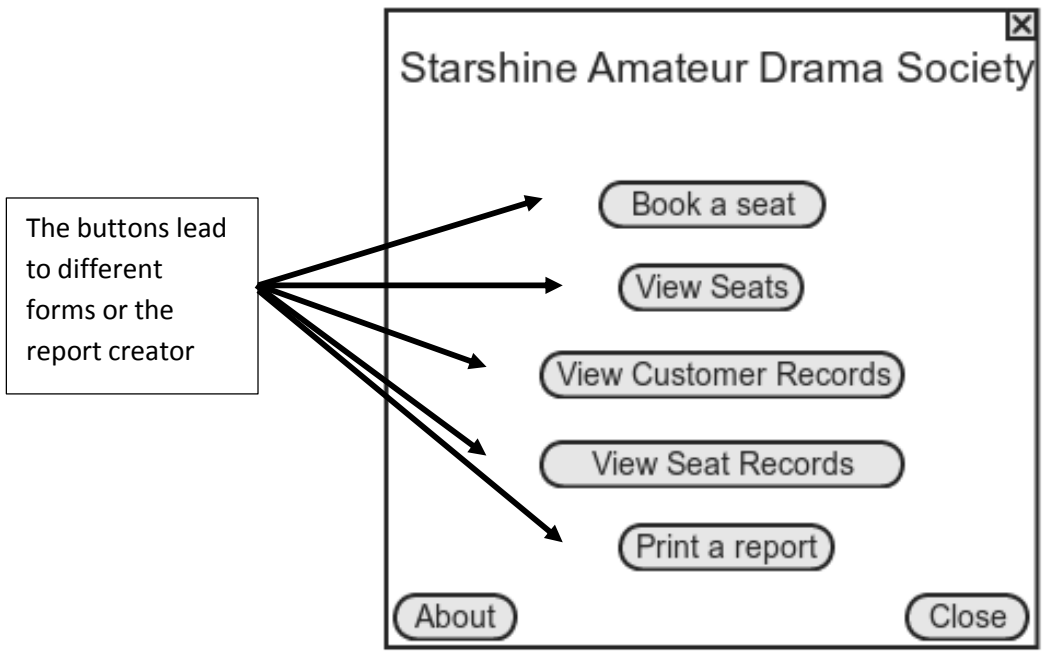
### User Interface

The user interface will be very professional as this will help the user navigate and use the program more easily. I have ensured that the functions of each button are easily shown by the text on the button and that the layout is simple and easy to navigate. The button which closes each form is always in the same position in each form, the bottom right hand corner. This also speeds up use of the program as the user will know where the buttons are.

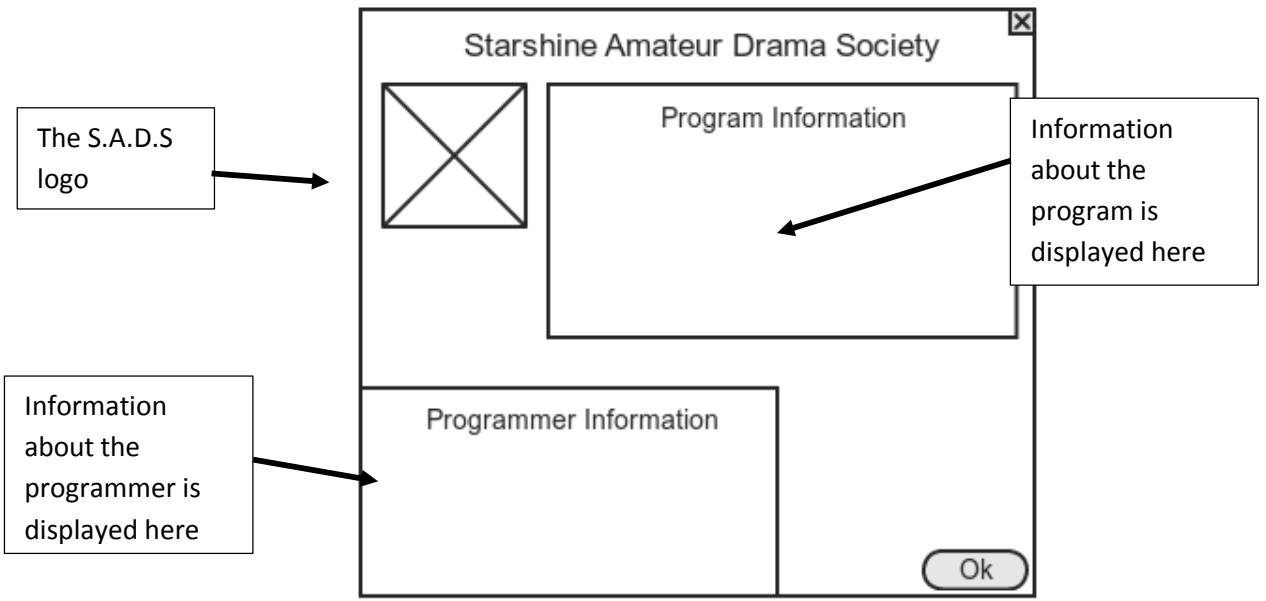
### Login Form



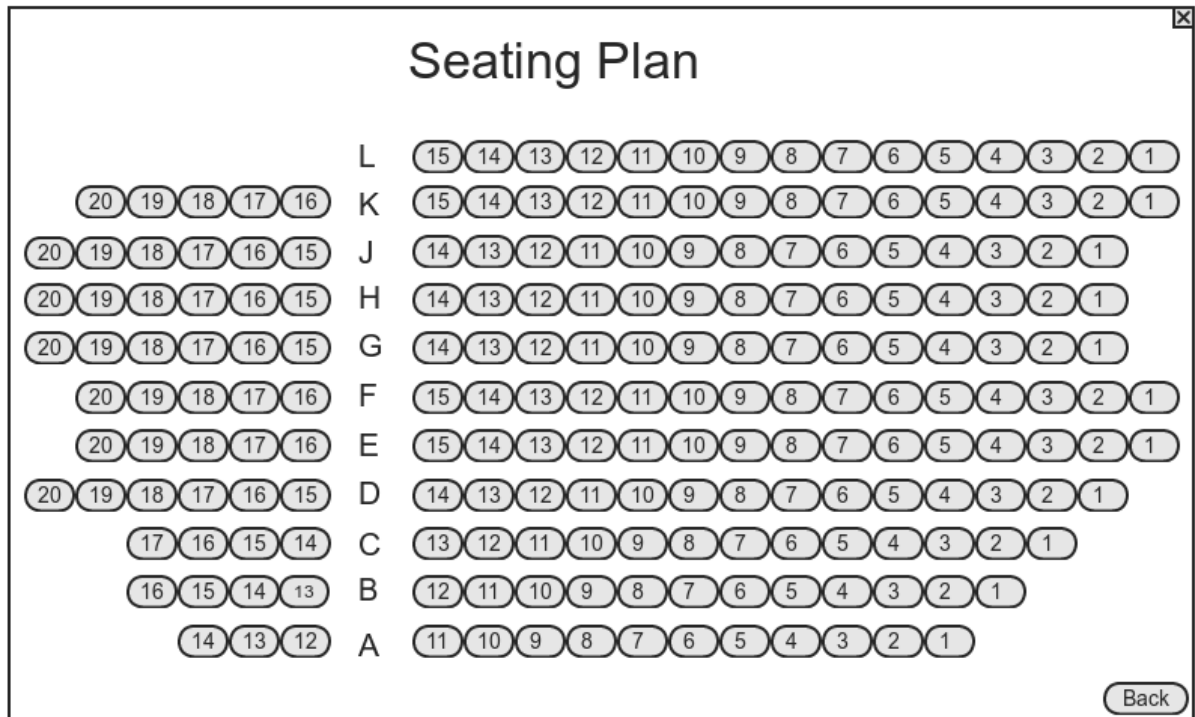
**Main Form**



**About Form**



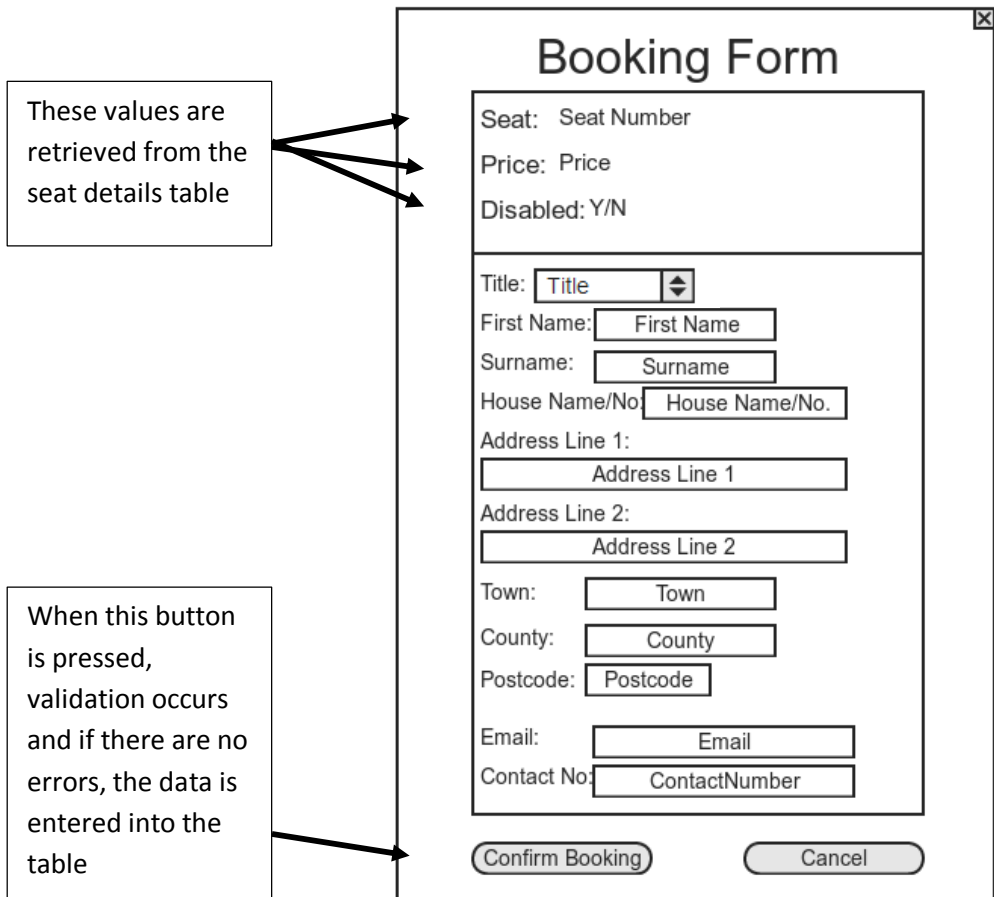
### Seating Plan



This is the seating plan of the theatre. Each seat is a button

Clicking on a seat brings you to the booking page for that particular seat

### Booking Form



### Customer Details Data view

Customer ID	Seat ID	Title	First Name	Surname	Date	House Name/No.	Address 1

Sorting

- Customer ID
- Seat ID
- Surname

Add Record Edit Record Delete Record Print Report

Back

The search function searches the entire table for the entered value

When a radio button is checked, the sorting is carried out

Opens the report creator window and allows the user to print the report

### Report view

Customer ID	Seat ID	Title	First Name	Surname

Print Back

Prints the report



### Seat Details Data view

The screenshot shows a window titled "Seat Details" with a search bar and a table. The table has four columns: "Customer ID", "Seat ID", "Price", and "Disabled". Below the table is a sorting menu with "Customer ID" selected and "Seat ID" as an option. There are also buttons for "Add Record", "Edit Record", "Delete Record", and "Back".

This form is almost identical to the Customer Details Data view

### Hardware and Software Requirements

#### For coding and maintaining the system

These requirements are those for using Visual studio 2012 which will be used for creating the system.

- Processor: 1.6GHz
- Operating System: Windows 7 SP1 or newer  
Windows Server 2008 R2 SP1 or newer
- Required Memory: 1Gb
- Required Hard Disk Space: 10 Gb
- Video Card: Capable of running at 1024x768 or higher resolution
- A monitor
- A mouse, keyboard and printer

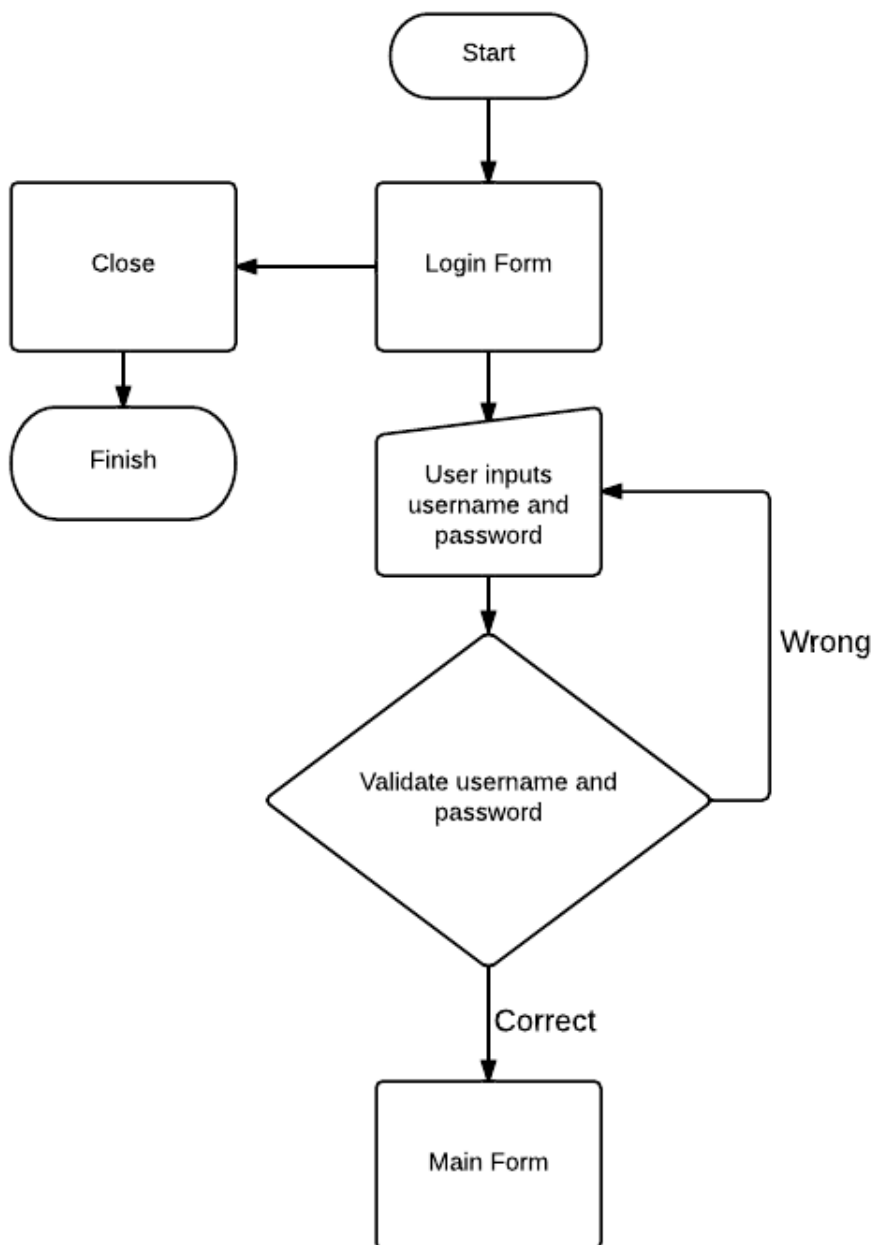
#### For using the system

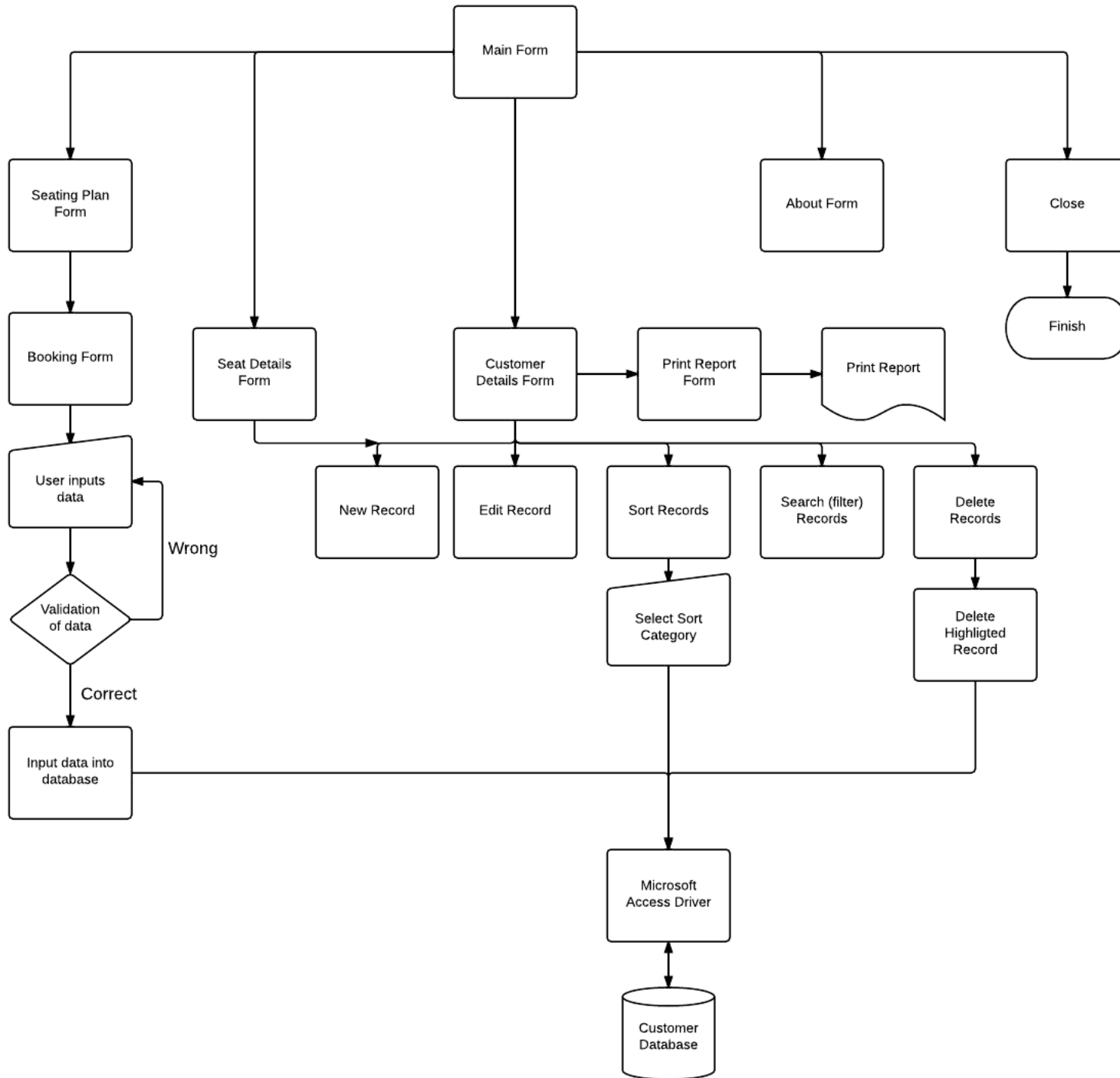
Using the system will require far less in terms of system requirements than for coding and maintaining the system as the system will be an exe file and relatively small.

- A monitor
- A mouse, keyboard and printer
- Processor: 1.6GHz
- Required Memory: 256Kb
- Required Hard Disk Space: 10Gb

## Processing Stages

### System Flowcharts





Huw Talliss

## Pseudocode

### LoginForm

```
IF username = admin AND password = password THEN
    Show MainForm
ELSE
    Show error message "Incorrect username or password"
ENDIF
```

### MainForm

```
IF About button is clicked THEN
    Show AboutForm
    Hide MainForm
ENDIF
```

```
IF Close button is clicked THEN
    End Program
ENDIF
```

### Choose Seat

```
IF seat button is clicked THEN
    day = currently selected tab
    seat = seat button clicked
    IF Day1Tab is currently selected THEN
        row = Get seat row from table of selected day
        IF the CustomerID of row is not 0 THEN
            Show messagebox asking if the user wants to delete the booking
            IF result = true THEN
                Show message box asking if the user is sure
                IF result = true THEN
                    Set the CustomerID of row to 0
                    Update the table
                    IF the CustomerID of row is 0 THEN
                        Show message saying that the booking has been deleted
                    ELSE
                        Show message saying that an error has occurred
                    ENDIF
                ENDIF
            ENDIF
        ENDIF
    ENDIF
ELSE
    Show DataEntry form
ENDIF
ENDIF
```

Huw Talliss

### **Tooltip**

```
IF the user mouses over a button THEN
    Get seat row from the table
    Show tooltip over button
ENDIF
```

### **Apply Colours**

```
FOR EACH control on tab
    IF control is a button THEN
        row = Get seat row from table
        IF the CustomerID of row is 0 THEN
            IF Disabled of row is true THEN
                Set button colour to Wheat
            ELSE
                Set button colour to White
            ENDIF
        ELSE
            IF Disabled of row is true THEN
                Set button colour to OrangeRed
            ELSE
                Set button colour to Tomato
            ENDIF
        ENDIF
    ENDIF
ENDIF
```

### **Book**

```
Row = Create new Customer Table
Assign Textbox data to Row
IF Database connection is closed THEN
    Open database connection
ENDIF
Create new table row using Row
ID = Get autotomatically incremented CustomerID
IF Database connection is open THEN
    Close database connection
ENDIF
IF Currently selected day is 1 THEN
    Get seat row from table
    Set the CustomerID of seat row to ID
    Update table
    Show message confirming the booking
    Close DataEntry form
```

Huw Talliss

## Data Entry validation

IF all text entry fields pass validation THEN

    Enable Book button

    Set Validation label to ""

ELSE

    Disable Book button

    Set Validation label to "One or more fields contain incorrect data"

IF focus on a text box is lost THEN

    IF the colour of the textbox is White OR GreenYellow OR the textbox is empty THEN

        Set colour of textbox to white

    ELSE

        Set colour of textbox to OrangeRed

    ENDIF

ENDIF

IF text of textbox is changed THEN

    IF textboxes data passes validation THEN

        Set the textbox's colour to GreenYellow

    ELSE

        Set the textbox's colour to OrangeRed

        Run BookBtnEnable

    ENDIF

ENDIF

## Searching

field = currently selected item in the Search combobox

filter = ""

token = text box's text

IF field is CustomerID THEN

    IF token passes validation THEN

        IF token is empty THEN

            Set filter to ""

        ELSE

            Set filter to "(" + field + " = " + token + ")"

        ENDIF

    ENDIF

ELSE

    IF token is empty THEN

        Set filter to ""

    ELSE

        Set filter to "(" + field + " LIKE '%" + token + "%'"

Apply the filter

Huw Talliss

## Evaluation Criteria

### Usability

I shall use a questionnaire to ascertain how user friendly my solution is and how easy it is to navigate and use by users of various skill levels from both my computing class and from the rest of my school.

The questionnaire will contain several questions for which the usability of the system will be rated against from 1-5.

I shall consider the usability of the system to be acceptable if most of the questions get a 4 or 5 rating.

### Suitability

The suitability of the solution will be analysed based upon if the program has successfully reached the main aims that I had set for the project. If the system does not fulfil the aims, then the suitability of the system is not successful.

I shall consider the system suitable if it fulfils all the aims I had set out for it.

### Performance

I will run several tests on many different aspects of the solution to see whether the program functions as intended and does not break when used.

If no major parts of the system fail the testing process or are very simply fixed, I will consider the performance of the system successful.

### Usability Questionnaire

Task	1 (Very hard)	2 (Difficult)	3 (Average difficulty)	4 (Easy)	5 (Very easy)
Navigation					
Booking a seat					
Deleting a booking					
Sorting records					
Searching for a record					
Printing a report					