

System Processes

An overview of the processes / functions of the system:

Login Form: Welcomes the user and allows them to login to their account.

Register Form: Allows user to register a new Artist account.

Register Staff Form: Allows staff members to register new staff accounts.

Add Booking Form: Allows users to create bookings on the system.

Artist Control Panel: Greets artist accounts after login, shows them all the functions they can use.

Staff Control Panel: Greets staff accounts after login, shows them all the functions they can use.

Booking Info Form: Allows users to view their bookings and generate a report / receipt if desired.

Booking Info (Staff) Form: Allows staff to view all bookings on the system and generate a report / receipt if desired.

Customer Info Form: Allows staff to view all customers registered to the system and allows them to use customer account manipulation functions.

Modify Booking Form: Allows users to select a booking to modify it then modify it.

Modify User Form: Allows staff members to select an artist account and modify it.

Support Form: Allows users to view support options for the community centre.

UserLog Form: Allows staff to view the log of all actions taken on the system.

Booking Invoice Report: Shows all relevant info to a booking that has been made.

Artists File

Name of file: Artists

Description: The file will contain information on each customer/artist that is currently registered on the database.

Expected number of users: I expect less than 200 registrations for the most part, however this could be exceeded if the business grows. I highly doubt the database will ever exceed 1000 fields.

Access Method: I will use direct access so that the system can search for, add, remove and modify fields.

Name of field	Type of field	Description of field	Size of field	Example of data
ArtistID	Autonumber, Primary Key	The automatically generated ID of the artist.	4	23
ActName	String	The name of the act or artist.	20	Rush
ManagerTitle	String	The title of the manager of the account.	5	Mr.
ManagerName	String	The name of the manager of the account.	20	John Smith
Username	String	The username of the account.	15	JohnS1998
Password	String	The password of the account.	15	John123
EmailAddress	String	The Email Address associated with the account.	40	JohnSmith@googlemail.com
Members	Number	The amount of members in the act.	2	4
U18	Boolean	Whether or not the majority of band members are under 18.	1	<input checked="" type="checkbox"/>
CardNumber	String	The 16 digit number on the payment card.	16	1234 1234 1234 1234
CardCCV	Integer	The 3 digit security code on the back of the payment card.	3	120
CardExpiry	Date	The date of expiry of the payment card.	5	11/25

Staff File

Name of file: Staff

Description: The file will contain information on each staff member that is currently registered on the database.

Expected number of users: I expect less than 150 registrations. It is unlikely that the business would hire more than this as the office space couldn't physically accommodate such a huge number of staff. Unless each person works once every two weeks (highly unlikely).

Access Method: I will use direct access so that the system can search for, add, remove and modify fields.

Name of field	Type of field	Description of field	Size of field	Example of data
StaffID	Autonumber, Primary Key	The automatically generated ID of the staff member.	4	36
Title	String	The title of the staff member.	5	Mrs.
Forename	String	The forename of the staff member.	15	Joanne
Surname	String	The surname of the staff member	15	Smith
EmailAddress	String	The email address of the staff member.	40	JoSmith@gmail.com
DateOfBirth	String	The date of birth of the staff member.	10	11/04/1995
Username	String	The username of the staff member's account.	15	JoSmith123
Password	String	The password of the staff member's account.	15	JoJo12

Bookings File

Name of file: Bookings

Description: The file will contain information on each booking that is registered on the database.

Expected number of users: I expect a large amount of bookings. Old bookings will remain in the system for archiving and invoicing purposes, therefore they may stack up quickly. This could mean thousands of bookings can potentially happen per year.

Access Method: I will use direct access so that the system can search for, add, remove and modify fields.

Name of field	Type of field	Description of field	Size of field	Example of data
BookingID	Autonumber, Primary Key	The automatically generated ID of the booking.	4	35
BookingStart	Date/Time	The time at which the booking starts.	10	11:00:00
BookingEnd	Date/Time	The time at which the booking ends.	10	2:00:00
EquipmentLend	String	The equipment that the artist wishes to borrow for the session.	50	2 Guitars, 2 Microphones.
BookingDate	Date/Time	The date the booking is scheduled for.	10	11/12/2017
Area	String	The area of the community centre that is being booked.	20	Practice Room 1
ArtistName	String	The name of the artist booking the space.	20	The John Smith Collective
TotalCost	Real	The total cost of the booking.	6	£10.00
EquipmentPrice	Real	The additional price of the equipment being borrowed.	6	£02.00
BasePrice	String	The category which decides on the rate at which the artists are charged.	40	Under 18 Discount
U18	Boolean	Whether or not the artist is under 18.	1	Yes
ArtistID	Integer, Foreign Key	The ArtistID from the Artist table.	4	21

User Log File

Name of file: UserActivityLog

Description: The file will contain information on every function activated in the database.

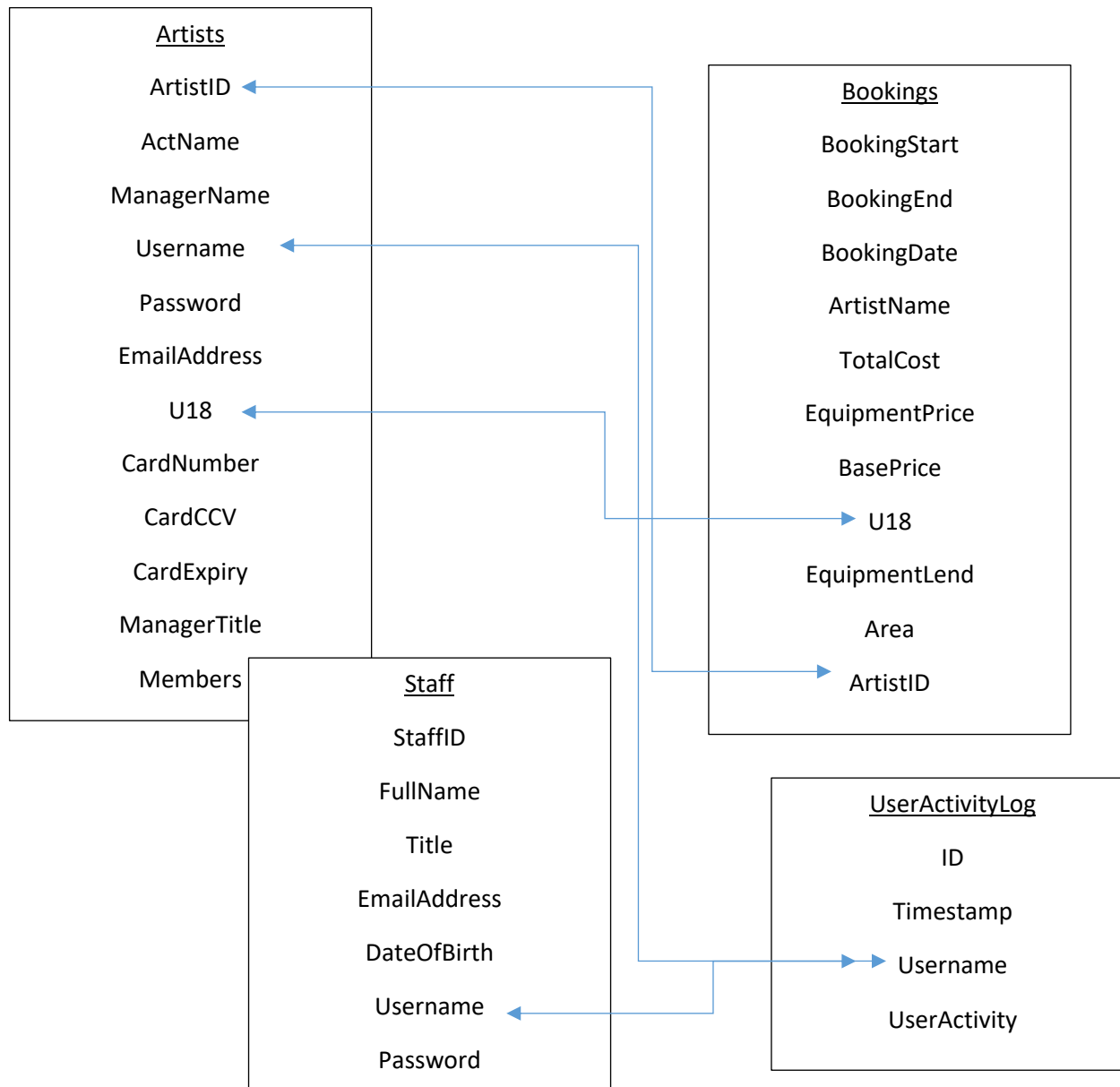
Expected number of users: I expect a huge amount of data to be stored in the table. Removal of data will be at the admin's discretion, but this serves as an archive of everything that happens on the system.

Access Method: I will use direct access so that the system can search for, add, remove and modify fields.

Name of field	Type of field	Description of field	Size of field	Example of data
ID	Autonumber, Primary Key	The automatically generated ID of the Activity.	50	{0C92DEB9-F6F6-4027-92D8-6548BF7EAA20}
Timestamp	Date/Time	The time at which the function occurs	20	16/03/2018 11:50:46
Username	String	The username of the user who performed the function.	20	Liam6
UserActivity	String	Brief description of what function was used.	35	"Logged In"

Entity Relationship Diagram

I have used a relationship where the Artist ID appears as a foreign key in the bookings table so that I can create a query based on this. This increases the security of the project as well as allowing for more extensive table queries.



Validation

There are some fields in each file which will need to be validated in order to make sure data is saved in an accurate and standardised way. This will make data easier to manipulate and display if needed.

Presence Check: Each required field will have a presence check in order to make sure that something has been entered into that field when new data is added into the system or old data is modified. A message box with an error message will be shown to the user if they do not fill in a required field.

CODE EXAMPLE :

```
If IsNull(Me.UsernameBox.Value) Or IsNull(Me.PasswordBox.Value) Then
```

```
    MsgBox ("You need to type a username and password!")
```

```
Exit Sub
```

```
End If
```

Format Check: Format checks will be useful on dates (00/00/0000), currencies (£00.00), email addresses (00000000@00000000.0000) and card numbers (0000 0000 0000 0000). Users will be able to type or select data and the format will automatically be applied. This is to be done within the program (object properties)

Length Check: Some fields will need to be checked as a correct length. Examples of these are:

- Card numbers (16 characters not including spaces, 20 with spaces.)
- CVV Codes (Must be 3 characters)
- Most personal data fields such as names, passwords, usernames etc. will need to be 4 or more characters.

Users will be notified if data is entered which isn't of a sufficient length.

CODE EXAMPLE :

```
If Len(phoneNum)<12 Then
```

```
    MsgBox 'Invalid phone number'
```

```
Exit Sub
```


End If

Type Check: A check to make sure the right type of data is entered, e.g. to make sure a username isn't accidentally entered into the card number field. Users will be notified if the wrong type of data is entered into a field.

UI Design

Colour Scheme: Purple and Blue

This page allows users to log in with a unique identifier or register for the system. These details are / will be saved in the tables in the database. This aims to reduce any user info being stored on paper, as everything is handled on the backend. Users simply need to remember their username and password and they will be taken to the main control panel.

Info Boxes

Login Screen

Login Page

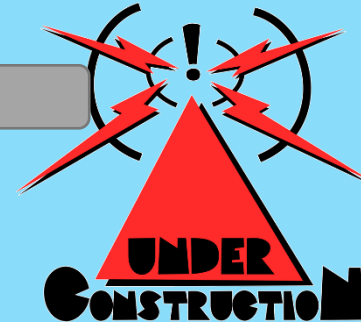
Label and Logo Image

Simple Labels

Text Boxes

Username

Password



Clearly Labelled Buttons

Login

Register

Login and Register Commands

This form will allow users to easily sign up so that they are ready to use the system straight away. It is laid out so that the system can easily collect all of the required and relevant information.

Register Artist

Back & Close Buttons by Title



Register User



Act Name

Manager Name

Manager Title



Intuitive / Time-Saving Drop Down Menu
(Combo box:
"Mr.;" "Mrs.;" "Miss.;" "Sir.;" "Dr.")

Form to Fill in

Username

Password

Password Confirm

Text Boxes with Labels

Email

No. Of Band Members

Intuitive / Time-Saving Checkbox

Under 18?

☒

Add User

Clearly Labelled Button – Add User Command

Optional Card Section

Card No.

Card Expiry

CCV

Register Staff

Back & Close Buttons by Title

Text Boxes with Labels



Register Staff



Staff Full Name

Staff Title



Username

Password

Confirm Password

Email Address

Date of Birth

Add Staff

Intuitive / Time-Saving Drop Down Box
(Combo Box:
"Mr."; "Mrs."; "Miss."; "Sir."; "Dr.")

This form will allow staff members to easily register other staff members so that they can use the system straight away.

Clearly Labelled Button – Command Button: Add Staff

Create Booking

Back & Close Buttons by Title

Create Booking

This form will allow new bookings to be added to the system and potentially paid for. This will collect all of the required information and will prevent clashes in the bookings from occurring.

Text Boxes with Labels

Intuitive / Time-Saving Drop Down Menus
(Combo Box: times 06:00-10:00)

Automatic Price Calculation

Intuitive / Time-Saving List Box

Intuitive / Time-Saving Calendar

Intuitive / Time-Saving Drop Down Menu
(Combo Box: "Practice Room 1"; "Practice Room 2"; "Main Hall"; "Studio")

Clearly Labelled Button

Add Booking

Act Name

Booking Start

Booking End

Equipment Needed

Equipment Price

What Equipment?

Date

Area

Price

This form will allow staff members to modify users by user request. Due to the modification of sensitive data, only staff members should be able to use this form.

Modify Artist

Back & Close Buttons by Title

Text Boxes with Labels

Intuitive / Time-Saving Drop Down Menu
(Combo Box:
"Mr."; "Mrs."; "Miss."; "Sir."; "Dr.")

Form to Fill in

Clearly Labelled Button –
Modify User Command

Modify User

Intuitive / Time-Saving Checkbox

List Containing Users registered to system.



Modify User



Act Name

Manager
Name

Manager
Title



Username

Password

Password
Confirm

Email

No. Of Band
Members

Under 18?

☒

Card No.

Card Expiry

CCV

Optional Card Section

View Customer Details

This form will allow staff members to view all of the users in the database. They are also given manipulation options for each user.

Back & Close Buttons by Title



View Users



Easy-to-view list of Customers

Clearly Labelled Button – Add A User Command

Add A User

Self-Explanatory Buttons

Clearly Labelled Button –
Modify A User Command

Modify A User

Modify Bookings

This form will allow users to choose a booking of theirs to modify. All bookings in the system will be displayed if it is made by them. These will be displayed in order of most recent first.

Back & Close Buttons by Title



Select Booking



Easy-to-view list of bookings

Click booking to modify

Modify Booking

Create Booking

Back & Close Buttons by Title

This form will automatically load with all of the data from the selected booking. Users can edit data and save the booking.

←

Modify Booking

×

Booking Start

Booking End

Text Boxes with Labels

Intuitive / Time-Saving Drop Down Menus

Equipment Needed

Intuitive / Time-Saving List Box

Automatic Price Calculation

Equipment Price

What Equipment?

Intuitive / Time-Saving Calendar

Date

Area

Price

Clearly Labelled Button

Modify Booking

This form will show users all of the bookings they have made. They are able to choose a booking and generate a report from it.

[View Bookings](#)

Back & Close Buttons by Title



View Bookings



Easy-to-view list of bookings

Click booking and generate invoice

--

Generate Invoice

Artist Control Panel



Create A Booking

Modify Booking

Show Invoice

Delete Account

Contact Support

Log Out

This is a complete navigation for Artists. Each function is accessed through this panel. Buttons are self explanatory.

This is a complete navigation for Staff.
Each function is accessed through this
panel. Buttons are self explanatory.

Staff Control Panel



View Artists

View Bookings /
Generate Invoice

View Artist Control
Panel (Debug)

Delete Account

Log Out

Add Staff

User Log

Back & Close Buttons by Title



Phone : 02920 121341

This is a support form where the information about the business is displayed to users who need support.

Email :
Support@Cathays.com

OPEN 6am-10pm

User Log

This form displays all of the user activity on the system along with their username, most recent first. Staff can view this for security and debugging reasons.

User Log

Easy-to-view list of activities

Back & Close Buttons by Title

Title explaining that this is an invoice.

Booking Invoice

Each relevant piece of information will be filled in automatically.

{Act Name}

System will automatically put users Artist Name here for identification purposes.

Booking ID

Base Price
Category

Duration

Date

Area

Equipment
Borrowed



Logo for brand purposes, especially important as this may be printed.

This form is automatically filled in when the user tries to generate a report on any given booking. This acts as a sort of receipt for this.

Total Cost

Paid?

☐

Checkbox allows user to keep track of payments.

Data Structures in User Files

Artists:

ArtistID

ActName

ManagerName

Username

Password

EmailAddress

U18

CardNumber

CardCCV

CardExpiry

ManagerTitle

Members

End

Bookings:

BookingID

BookingStart

BookingEnd

BookingDate
ArtistName
TotalCost
EquipmentPrice
BasePrice
U18
EquipmentLend
Area
ArtistID

End

Staff:

StaffID
FullName
Title
EmailAddress
DateOfBirth
Username
Password

End

UserActivityLog:

ID

Timestamp

Username

UserActivity

End

Forms

Login:

Begin Prog:

Form load:

If close button is clicked:

Close the login form.

End sub

If Login button is clicked:

Connect to database and tables for Artists and Staff.

If either username or password boxes are empty, prompt user to type username and password using messagebox.

Loop through both tables until instance of matching username and password are found.

If username and password are not found, show messagebox saying so.

If username and password found in Artists table, save username as a global variable and enter Artist main menu.

If username and password found in Staff table, save username as global variable and enter Staff main menu.

Close connections to tables to avoid corruption.

Make an entry in the Userlog that someone has logged in.

Close login form.

End Sub

If Register button is clicked:

Connect to database and staff table.

Loop through staff table until instance of matching username and password found.

If staff login is found, go to add staff form.

If no staff login found, go to add user form.

Close connection to staff table to avoid corruption.

End Sub

End Prog

AMain (Artist Main Menu):

Begin Prog:

Form Load

If close button is clicked:

Close Artist main menu.

End Sub

If Add Booking button is clicked:

Open Add Booking Form.

End Sub.

If Modify Booking button is clicked:

Open Modify Booking Form

End Sub

If Delete Account button is clicked:

Create variable where 'DeleteVal' is a box asking the user to input DELETE to confirm deletion of account.

If user types DELETE and confirms, their account will be deleted with a messagebox confirming this. the Login form will open and the Artist menu will close.

If the user fails to type the confirmation word or exits the delete box, a messagebox will confirm the account has not been deleted.

End Sub

If Contact Support button is clicked:

Open Support form.

End Sub

If Log Out button is clicked:

Open Login Form, Close Artist main menu.

Add to UserLog to say that the user has logged out.

End Sub

If Show Invoice button is clicked:

Open BookingInfo Form

End Sub

End Prog

SMain (Staff Main Menu):

Begin Prog

Form Load

If Close button is clicked:

Close Staff main menu form.

End Sub

If View Artists button is clicked:

Open Customer Form.

End Sub

If Delete Account button is clicked.

Create variable where 'DeleteVal' is a box asking the user to input DELETE to confirm deletion of account.

If user types DELETE and confirms, their account will be deleted with a messagebox confirming this. the Login form will open and the Staff menu will close.

If the user fails to type the confirmation word or exits the delete box, a messagebox will confirm the account has not been deleted.

End Sub

If Add Staff button is clicked:

Open Add Staff form

End Sub

If Artist Control Panel Debug button is clicked:

Open Artist Main Menu

End Sub

If Log Out button is clicked:

Open Login form and close Staff main menu.

End Sub

If UserLog button is clicked:

Open UserLog Form

End Sub

End Prog

AddBooking:

Begin Prog

Form Load

If Close button is clicked:

If the user presses the (X) button in the top right corner, it will close the current form.

End Sub

If Create Booking button is clicked:

If any of the required boxes are empty, display an error message.

Create variables called db, table and table2. Variable as a Database, and 2 Recordsets Respectively.

Connect variable db to the current database

Connect variable table to the Bookings table

Connect variable table2 to the Artists table

Find the first record in the Artists table where the CardCCV field in the table is equal to the CCV value in the textbox.

Find the first record in the Artists table where the CardNumber field in the table is equal to the Card Number value in the textbox.

Define the varX variable and use a Dlookup to find the actname of the user based on the username.

Define the varY variable and use a Dlookup to find the Artist ID of the user based on the username.

Define the varZ variable and use a Dlookup to find the Booking Start based on the area and the date that the user put in.

Define the varA variable and use a Dlookup to find the Booking End based on the area and the date that the user put in.

Check whether the time, date and area overlap with another booking. If so, the booking will not be made. This is a way of avoiding double bookings.

Add all relevant booking info to new record in table.

Display confirmation message box.

Create record in userlog to say user created a booking.

Close Add Booking form.

End Sub

If Back button is clicked:

If the user presses the (<-) button in the top right corner, it will close the current form.

End Sub

If Equipment Price box is clicked:

Calculate and display preset prices based on what list element is clicked.

End Sub

If Total Price box is clicked:

Show messagebox prompting user to fill in all fields if not done already.

Connect to artists table to get data.

Calculate and display price of booking based on equipment and user's discount eligibility.

End Sub

End Prog

Add Staff Form:

Begin Prog

Form Load

If Back button is clicked:

Close Add Staff Form

End Sub

If Register button is clicked:

Show message box prompting user to fill in all starred fields if not already done so.

Show message box notifying user of mismatch if passwords do not match.

Connect to staff table.

Add all info to staff table.

Show msgbox confirming staff registration.

Open Staff main menu

Add Staff registration to UserLog

Close Add Staff Form.

End Sub

If Close button is clicked:

Open Login Form

Close Add Staff Form.

End Sub

End Prog

Add User:

Begin Prog

Form Load

If the Close button is clicked:

Close AddUser Form.

If the Register button is clicked:

Show message box prompting user to fill in all starred fields if not already done so.

Show message box notifying user of mismatch if passwords do not match.

Connect to Artists Table

Add all info to Artists table.

Show msgbox confirming Artist registration.

Close Add User Form

Open Login Form

End Sub

If Back button is clicked:

Open Login Form

Close Add User Form

End Sub

End Prog

BookingInfo (List is filtered to only show user bookings):

Begin Prog

Form Load

If Close button is clicked:

Close Booking Info Form

End Sub

If Show Invoice is clicked:

Connect to bookings table

Save BookingID number in box as bookid

Find first invoice where the BookingID matches bookid

If there's no match, show message box telling user.

Define the report name

Make a copy of the report template and input relevant details

Save and open report

Close table connection to prevent corruption

Add invoice generation to UserLog.

Close BookingInfo Form

End Sub

If Back button is clicked:

Close Booking Info form

End Sub

End Prog

BookingInfoStaff:

Begin Prog

Form Load

If Close button is clicked:

Close Booking Info Form

End Sub

If Show Invoice is clicked:

Connect to bookings table

Save BookingID number in box as bookid

Find first invoice where the BookingID matches bookid

If there's no match, show message box telling user.

Define the report name

Make a copy of the report template and input relevant details

Save and open report

Close table connection to prevent corruption

Add invoice generation to UserLog.

Close BookingInfo Form

End Sub

If Back button is clicked:

Close Booking Info form

End Sub

End Prog

Customer Form:

Begin Prog

Form Load

If Close button is clicked:

Close Customer Form

End Sub

If Add New Customer button is clicked:

Open Add User Form

End Sub

If Modify Customer button is clicked:

Open Modify User Form

Close Customer Form

End Sub

If Back button is clicked:

Close Customer Form

End Sub

End Prog

ModifyBooking:

Begin Prog

Form Load

If Close button is clicked:

Close Modify Booking form

End Sub

If Modify button is clicked:

Connect to database and Bookings table

Save ID of booking that user chooses

Find the relevant data for the chosen booking in the table based on the booking ID

Close ModifyBooking, Open ModifyBooking_p2

End Sub

If Back button is clicked:

Close ModifyBooking

Open Artist Main Menu

End Sub

End Prog

ModifyBooking_p2:

Form Load:

Connect to booking table

Fill in the relevant data automatically based on what was selected in the previous form.

End Sub

If Close button is clicked:

Close ModifyBooking_p2 form

If Modify button is clicked:

Connect to bookings table

Replace data in relevant record with newly input data

Show messagebox confirming success

Make entry into UserLog about modified booking

Close ModifyBooking_p2

Open Artist Main Menu

End Sub

If Back button is clicked:

Close ModifyBooking_p2

Open ModifyBooking

End Sub

If Total Price box is clicked:

Calculate and display price of booking based on equipment and user's discount eligibility.

End Sub

If Equipment Price box is clicked:

Calculate and display preset prices based on what list element is clicked

End Sub

End Prog

ModifyUser:

Begin Prog

Form Load:

Set FoundCustomer as False

End Sub

If Close button is clicked:

Close ModifyUser Form

End Sub

If Back button is clicked:

Open CustomerForm

Close Modify User Form

End Sub

If Find button is clicked:

Connect to Artists table

If ID is invalid, ask for valid ID in messagebox

Find first instance of matching ID in the Artists table

If none match, say so in messagebox.

Fill in the form's fields with all relative info automatically.

Set FoundCustomer to true

End Sub

If Modify Button is clicked:

If user hasn't filled in any of the starred fields, tell them this in messagebox.

Connect to artists table

Ask for user ID if not entered.

Update the record with changed information

Close the Artists table to prevent corruption

Confirm successful modificaton with messagebox

Open Staff Main Menu and add user modify to UserLog

Close ModifyUser form.

End Sub

End Prog

Support Form:

Begin Prog

Form Load

If Close button is clicked:

Close support form.

End Sub

End prog

UserLog Form:

Begin prog

Form Load

If Close button is clicked:

Close UserLog form

End Sub

If Back button is clicked:

Close UserLog form

End Sub

End prog

Interconnectivity

I will now detail which forms access which tables.

FORMAT:

FORM:

- FILE THAT FORM ACCESSES

Login:

- Artists
- Staff
- UserActivityLog

AddBooking:

- Bookings
- Artists
- UserActivityLog

AddStaff:

- Staff
- UserActivityLog

AddUser:

- Artists
- UserActivityLog

BookingInfo:

- Bookings

BookingInfoStaff:

- Bookings

CustomerForm:

- Artists

ModifyBooking:

- Bookings

ModifyBooking_p2:

- Bookings
- UserActivityLog

ModifyUser:

- Artists
- UserActivityLog

UserLog:

- UserActivityLog

AMain:

- Artists

SMain:

- Staff

Due to the rather large frequency of interconnectivity, it is important that I correctly program connections in order to avoid mass corruption in any parts of my database.

I shall now create 'Create, Read, Update, Delete' tables to further explain how each form will interact with my files.

Login	Create	Read	Update	Delete
Artists		X		
Bookings				
Staff		X		
UserActivityLog				

AddBooking	Create	Read	Update	Delete
Artists				
Bookings	X	X		
Staff				
UserActivityLog			X	

AddStaff	Create	Read	Update	Delete
Artists				
Bookings				
Staff	X	X		
UserActivityLog			X	

AddUser	Create	Read	Update	Delete
Artists	X	X		
Bookings				
Staff				
UserActivityLog			X	

BookingInfo	Create	Read	Update	Delete
Artists		X		
Bookings		X		
Staff				
UserActivityLog				

BookingInfoStaff	Create	Read	Update	Delete
Artists				
Bookings		X		
Staff				
UserActivityLog				

CustomerForm	Create	Read	Update	Delete
Artists		X		
Bookings				
Staff				
UserActivityLog				

ModifyBooking	Create	Read	Update	Delete
Artists		X		
Bookings		X		
Staff				
UserActivityLog				

ModifyBooking_p2	Create	Read	Update	Delete
Artists		X		
Bookings		X	X	
Staff				
UserActivityLog			X	

ModifyUser	Create	Read	Update	Delete
Artists		X	X	
Bookings				
Staff				
UserActivityLog			X	

UserLog	Create	Read	Update	Delete
Artists				
Bookings				
Staff				
UserActivityLog		X		

AMain	Create	Read	Update	Delete
Artists		X		X
Bookings				
Staff				
UserActivityLog				

SMain	Create	Read	Update	Delete
Artists				
Bookings				
Staff		X		X
UserActivityLog				