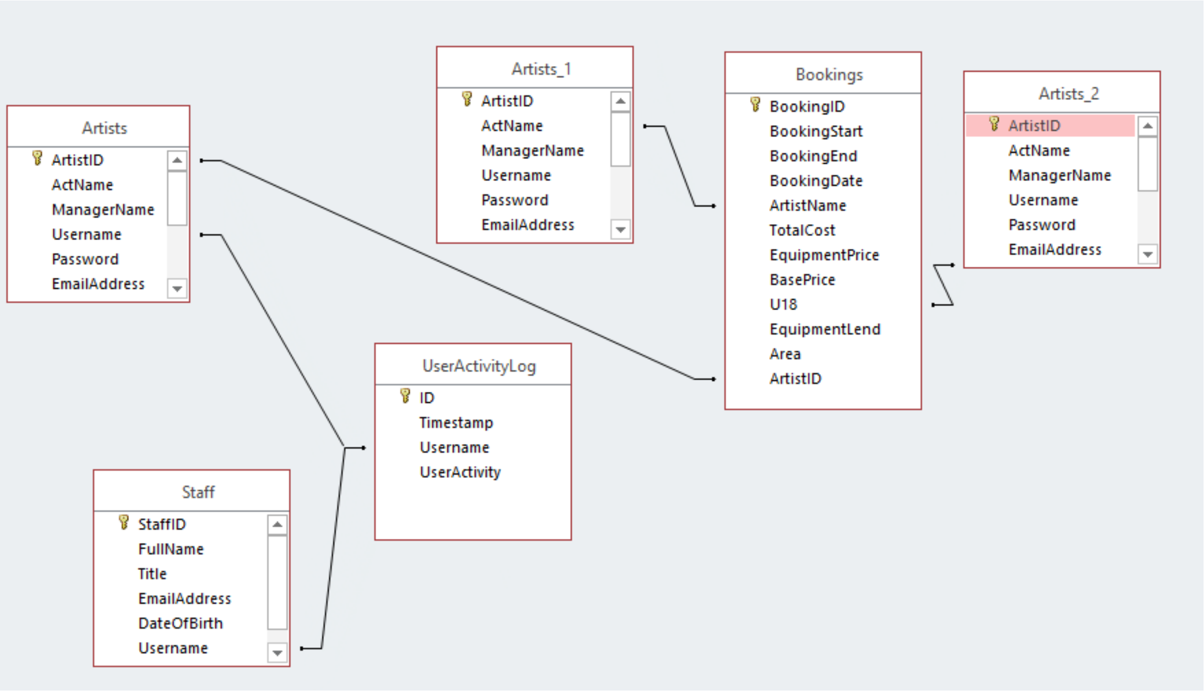
SOFTWARE DEVELOPMENT

Jack Browne - 555784

CARDIFF AND VALE COLLEGE



<u>Artists File</u>

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	☑
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"

Code behind each page

Login

```
Option Compare Database
Option Explicit



Private Sub Command11_Click()
    DoCmd.Close acForm, ("Login")
    'Close the login form.


End Sub

Public Sub LoginButton_Click()

    Dim db As DAO.Database
    Dim ATable As DAO.Recordset
    Dim STable As DAO.Recordset

    Set db = CurrentDb()
    Set ATable = db.OpenRecordset("Artists", dbOpenDynaset)
    Set STable = db.OpenRecordset("Staff", dbOpenDynaset)
    'Connect to database and tables for Artists and Staff.

    If IsNull(Me.UsernameBox.Value) Or IsNull(Me.PasswordBox.Value) Then
        MsgBox ("You need to type a username and password!")
        Exit Sub
    End If
    'If either username or password boxes are empty, prompt user to type username and password using messagebox.
```

```
   ATable.FindFirst ("Username = '" & Me.UsernameBox.Value & "' And Password = '" & Me.PasswordBox.Value & "'")

   STable.FindFirst ("Username = '" & Me.UsernameBox.Value & "' And Password = '" & Me.PasswordBox.Value & "'")

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


   If ATable.NoMatch And STable.NoMatch Then

      MsgBox ("User not found.")

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

   Else

      If ATable.NoMatch Then

         DoCmd.OpenForm ("SMain")

         globUsername = Me.UsernameBox

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

      Else


      globUsername = Me.UsernameBox

      DoCmd.OpenForm ("AMain")

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


      End If

   End If


ATable.Close

STable.Close


Globals.LoggingUsers "Logged In"


DoCmd.Close acForm, ("Login")


'Close connections to tables to avoid corruption.

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

```
'Close login form.



End Sub


Private Sub RegisterButton_Click()


Dim db As DAO.Database

Dim STable As DAO.Recordset


Set db = CurrentDb()

Set STable = db.OpenRecordset("Staff", dbOpenDynaset)

'Connect to database and staff table.


STable.FindFirst ("Username = '" & Me.UsernameBox.Value & "' And Password = '" &
Me.PasswordBox.Value & "'")

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


If STable.NoMatch Then

   DoCmd.OpenForm ("AddUser")

   DoCmd.Close acForm, ("Login")

Else

   DoCmd.OpenForm ("AddStaff")

   DoCmd.Close acForm, ("Login")

End If

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

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



STable.Close

'Close connection to staff table to avoid corruption.


End Sub
```

AMain

```
Option Compare Database

Private Sub Command11_Click()

   DoCmd.Close acForm, ("AMain")
   'Close Artist main menu.

End Sub

Private Sub Command2_Click()

   DoCmd.OpenForm ("AddBooking")
   'Open Add Booking Form.

End Sub


Public Sub Command3_Click()

   DoCmd.OpenForm ("ModifyBooking")
   'Open Modify Booking Form

End Sub

Private Sub Command5_Click()

   DeleteVal = InputBox("If you are certain you wish to delete your account, please type 'DELETE'")
   'Create variable where 'DeleteVal' is a box asking the user to input DELETE to confirm deletion of account.
   If (DeleteVal = "DELETE") Then
```

```
        MsgBox ("Account successfully deleted. Logging out now.")
        CurrentDb.Execute ("DELETE * FROM Artists WHERE Username = """ & globUsername & """")
        DoCmd.OpenForm ("Login")
        DoCmd.Close acForm, ("AMain")
        '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.
    Else
        MsgBox ("You did not type the verification keyword correctly. Account not deleted")
        '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 If


End Sub


Private Sub Command6_Click()


    DoCmd.OpenForm ("Support")
    'Open Support form.


End Sub


Private Sub Command7_Click()


    DoCmd.OpenForm ("Login")
    Globals.LoggingUsers "Logged Out"
    DoCmd.Close acForm, ("AMain")
    'Open Login Form, Close Artist main menu.
    'Add to UserLog to say that the user has logged out.


End Sub


Public Sub Command8_Click()
```

```
    globUsername = globUsername
    DoCmd.OpenForm ("BookingInfo")
    'Open BookingInfo Form

End Sub

Public Function Getuser()

    Getuser = globUsername

End Function
```

SMain

```
Option Compare Database

Private Sub Command11_Click()

    DoCmd.Close acForm, ("SMain")

End Sub

Private Sub Command2_Click()

    DoCmd.OpenForm ("CustomerForm")

End Sub


Public Sub Command3_Click()

    DoCmd.OpenForm ("BookingInfoStaff")
```

```
End Sub


Private Sub Command5_Click()


   DeleteVal = InputBox("If you are certain you wish to delete your account, please type 'DELETE'")

   If (DeleteVal = "DELETE") Then

      MsgBox ("Account successfully deleted. Logging out now.")

      CurrentDb.Execute ("DELETE * FROM Artists WHERE Username = '" & globUsername & "'")

      CurrentDb.Execute ("DELETE * FROM Staff WHERE Username = '" & globUsername & "'")

      DoCmd.OpenForm ("Login")

      DoCmd.Close acForm, ("AMain")

   Else

      MsgBox ("You did not type the verification keyword correctly. Account not deleted")

   End If


End Sub


Private Sub Command6_Click()


   DoCmd.OpenForm ("AddStaff")


End Sub


Private Sub Command7_Click()


   DoCmd.OpenForm ("Login")

   DoCmd.Close acForm, ("SMain")


End Sub


Public Sub Command8_Click()
```

```
    globUsername = globUsername
    DoCmd.OpenForm ("AMain")

End Sub

Public Function Getuser()

    Getuser = globUsername

End Function



Private Sub Command9_Click()

    DoCmd.OpenForm ("UserLog")

End Sub
```

AddBooking

```
Option Compare Database

Private Sub Close_Click()
'Create private sub that is activated upon clicking the (X) button. Aim: Close the page.

    DoCmd.Close acForm, ("AddBooking")
    'If the user presses the (X) button in the top right corner, it will close the current form.

End Sub

Public Sub CreateBtn_Click()
'Create private sub when Create Booking button is clicked. Aim: Create the booking
based on given info.
```

```
    If IsNull(ActName) Or IsNull(BookingStart) Or IsNull(BookingEnd) Or IsNull(Date) Or (Area.ListCount < 1) Or IsNull(CCV) Then

        MsgBox ("Sorry, you must fill in the starred boxes.")

    Exit Sub

    End If

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


    Dim db As DAO.Database

    Dim table As DAO.Recordset

    Dim table2 As DAO.Recordset

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


    Set db = CurrentDb()

    'Connect variable db to the current database

    Set table = db.OpenRecordset("Bookings", dbOpenDynaset)

    'Connect variable table to the Bookings table

    Set table2 = db.OpenRecordset("Artists", dbOpenDynaset)

    'Connect variable table2 to the Artists table

    table2.FindFirst ("CardCCV = '" & Me.CCV.Value & "'")

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

    table2.FindFirst ("CardNumber = '" & Me.CardNum.Value & "'")

    '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.


    Dim oldDate As Date

    Dim newDate As Date

    Dim Hour As Integer

    'Declaring some variables as Date and Integer Variables. These will make calculations a bit simpler to understand.

    Hour = newDate

    oldDate = Me.BookingStart.Value

    newDate = DateDiff("h", oldDate, Me.BookingEnd.Value)
```

```
   'Due to changes with the code, this is more of a 'patch-up' job to keep the calculations
working as intended.


   Dim varX As Variant

   varX = DLookup("[ActName]", _

   "Artists", "[Username] = '" & globUsername & "'")

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

   'The username is a global variable that was saved when the user logged in. This
increases security and intuity.


   Dim varY As Variant

   varY = DLookup("[ArtistID]", _

   "Artists", "[Username] = '" & globUsername & "'")

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

   'The username is a global variable that was saved when the user logged in. This
increases security and intuity.


   Dim varZ As Variant

   varZ = DLookup("[BookingStart]", _

   "Bookings", "CStr([Area]) = '" & CStr(Me.Area.Value) & "' And CStr([BookingDate]) = '"
& CStr(Me.Date.Value) & "'")

   '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.


   Dim varA As Variant

   varA = DLookup("[BookingEnd]", _

   "Bookings", "CStr([Area]) = '" & CStr(Me.Area.Value) & "' And CStr([BookingDate]) = '"
& CStr(Me.Date.Value) & "'")

   '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.



   If (varZ >= Me.BookingStart.Value And varZ <= newDate) Or (varA >=
Me.BookingStart.Value And varZ <= newDate) Then
```

```
        MsgBox ("Sorry, that time or area is already booked. Please try another time or
area.")

        'Checks 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.

    Else

        If (table2.NoMatch) Then

        MsgBox ("Incorrect card security code or card number.")

        Else

            table.AddNew

            table("BookingStart") = Me.BookingStart.Value

            table("BookingEnd") = Me.BookingEnd.Value

            table("EquipmentLend") = Me.Equip.Value

            table("BookingDate") = Me.Date.Value

            table("Area") = Me.Area.Value

            table("TotalCost") = Me.TotalPrice.Value

            table("EquipmentPrice") = Me.EquipmentPrice.Value

            table("U18") = Me.Check33.Value

            table("BasePrice") = "Self-Booked"

            table("ArtistName") = varX

            table("ArtistID") = varY

            table.Update

            table.Close

            MsgBox ("Booking was successful.")

            Globals.LoggingUsers "Created Booking"

            DoCmd.Close acForm, ("AddBooking")

            '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 If

    End If
End Sub


Private Sub Command6_Click()

    DoCmd.Close acForm, ("AddBooking")

End Sub


Private Sub Detail_Activate()


End Sub

Private Sub Equipment_Click()

End Sub

Private Sub PriceCat_Click()



End Sub

Private Sub EquipmentPrice_Click()


    If (Equipment.ListIndex = 0) Then
        With Me
            .EquipmentPrice.Text = 0
        End With
    End If
    If (Equipment.ListIndex = 1) Then
        With Me
            .EquipmentPrice.Text = 2
        End With
    End If
    If (Equipment.ListIndex = 2) Then
        With Me
            .EquipmentPrice.Text = 4
        End With
    End If
    If (Equipment.ListIndex = 3) Then
        With Me
            .EquipmentPrice.Text = 6
        End With
    End If
    If (Equipment.ListIndex = 4) Then
        With Me
            .EquipmentPrice.Text = 8
        End With
    End If
    If (Equipment.ListIndex = 5) Then
        With Me
            .EquipmentPrice.Text = 10
        End With
    End If

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

```
End Sub




Private Sub TotalPrice_Click()


If IsNull(ActName) Or IsNull(BookingStart) Or IsNull(Hrs) Then
   MsgBox ("Please Fill in Required Fields.")
Else

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

   Dim db As DAO.Database
   Dim table As DAO.Recordset

   Set db = CurrentDb()
   Set table = db.OpenRecordset("Artists", dbOpenDynaset)

   'Connect to artists table to get data.

   Hrs = DateDiff("h", Me.BookingStart.Value, Me.BookingEnd.Value)



If IsNull(Hrs) = False Then
   If (Forms!AddBooking!Check33 = True) Then
      With Me
         .TotalPrice.Text = EquipmentPrice + 6 + Hrs
      End With
   Else
      With Me
         .TotalPrice.Text = EquipmentPrice + (3 * Hrs) + Hrs
      End With
   'Calculate and display price of booking based on equipment and user's discount eligibility.

   End If
Else
MsgBox ("Please Fill in Required Fields.")
End If
End If
End Sub
```

AddStaff

```
Option Compare Database

Private Sub Command11_Click()

   DoCmd.Close acForm, ("AddStaff")
   'Close Add Staff Form

End Sub

Private Sub Command32_Click()

   If IsNull(Me.Act.Value) Or IsNull(Me.ManT.Value) Or IsNull(Me.Un.Value) Or IsNull(Me.pw.Value) Or IsNull(Me.Pw2.Value) Or IsNull(Me.Em.Value) Then
      MsgBox ("You must fill in the starred (*) fields.")
      Exit Sub
   End If

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

   If ((Me.pw.Value) <> (Me.Pw2.Value)) Then
      MsgBox ("The two passwords typed do not match.")
      Exit Sub
```

```
End If

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

Dim db As DAO.Database
Dim table As DAO.Recordset

Set db = CurrentDb()
Set table = db.OpenRecordset("Staff", dbOpenDynaset)

'Connect to staff table.

table.AddNew
table("Title") = Me.ManT.Value
table("FullName") = Me.Act.Value
table("Username") = Me.Un.Value
table("Password") = Me.pw.Value
table("EmailAddress") = Me.Em.Value
table("DateOfBirth") = Me.Mems.Value
table.Update

'Add all info to staff table.

table.Bookmark = table.LastModified
MsgBox ("New staff member created with ID " & table!StaffID & ".")
table.Close

DoCmd.OpenForm ("SMain")
Globals.LoggingUsers "Created Staff Login"

DoCmd.Close acForm, ("AddStaff")
```

```
End Sub

Private Sub Command6_Click()

    DoCmd.OpenForm ("Login")
    DoCmd.Close acForm, ("AddStaff")

    'Open Login Form
    'Close Add Staff Form.



End Sub
```

Add User

```
Option Compare Database

Private Sub Command11_Click()

    DoCmd.Close acForm, ("AddUser")
    'Close AddUser Form.

End Sub

Private Sub Command32_Click()

    If IsNull(Me.Act.Value) Or IsNull(Me.ManT.Value) Or IsNull(Me.ManN.Value) Or
IsNull(Me.Un.Value) Or IsNull(Me.pw.Value) Or IsNull(Me.Pw2.Value) Or
IsNull(Me.Em.Value) Then
        MsgBox ("You must fill in the starred (*) fields.")
        Exit Sub
    End If

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

```
If ((Me.pw.Value) <> (Me.Pw2.Value)) Then
    MsgBox ("The two passwords typed do not match.")
    Exit Sub
End If

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

Dim db As DAO.Database
Dim table As DAO.Recordset

Set db = CurrentDb()
Set table = db.OpenRecordset("Artists", dbOpenDynaset)

'Connect to Artists Table

table.AddNew
table("ActName") = Me.Act.Value
table("ManagerTitle") = Me.ManT.Value
table("ManagerName") = Me.ManN.Value
table("Username") = Me.Un.Value
table("Password") = Me.pw.Value
table("EmailAddress") = Me.Em.Value
table("Members") = Me.Mems.Value
table("U18") = Me.Check22.Value
table("CardNumber") = Me.Cardn.Value
table("CardCCV") = Me.CCV.Value
table("CardExpiry") = Me.Exp.Value
table.Update

'Add all info to Artists table.

table.Bookmark = table.LastModified
MsgBox ("New act created with ID " & table!ArtistID & ".")
```

```
    table.Close

    'Show msgbox confirming Artist registration.

    DoCmd.OpenForm ("Login")
    DoCmd.Close acForm, ("AddUser")

    'Close Add User Form
    'Open Login Form



End Sub

Private Sub Command6_Click()

    DoCmd.OpenForm ("Login")
    DoCmd.Close acForm, ("AddUser")

    'Open Login Form
    'Close Add User Form

End Sub
```

BookingInfo

```
Option Compare Database

Private Sub Command11_Click()

    DoCmd.Close acForm, ("BookingInfo")
    'Close Booking Info Form

End Sub
```

```
Public Sub Command5_Click()

    Dim db As DAO.Database
    Dim invoice As DAO.Recordset

    Set db = CurrentDb()
    Set invoice = db.OpenRecordset("Bookings", dbOpenDynaset)
    'Connect to bookings table

    If IsNull(Me.BookingIDBox.Value) Then
    MsgBox ("Please choose a booking to view.")
    Else


    bookid = Me.BookingIDBox.Value
    'Save BookingID number in box as bookid

    invoice.FindFirst ("BookingID = " & bookid)
    'Find first invoice where the BookingID matches bookid

    If invoice.NoMatch Then
        MsgBox ("No invoice with that Booking ID could be found")
        Exit Sub
    End If
    'If there's no match, show message box telling user.




    Dim RepName As String
    RepName = "Invoice - Booking #" & bookid & " InvID: " & Int((300000 - 1 + 1) * Rnd + 300000)
    DoCmd.CopyObject , RepName, acReport, "Invoice Template"
    DoCmd.OpenReport RepName, acViewDesign, , , acHidden
    Reports(RepName).RecordSource = "SELECT ArtistName, BookingID, BookingStart,
BookingEnd, BasePrice, BookingDate, Area, EquipmentLend, TotalCost FROM Bookings
WHERE BookingID =" & Me.BookingIDBox.Value

    DoCmd.Save acReport, RepName

    DoCmd.OpenReport RepName, acViewPreview

    invoice.Close


    Globals.LoggingUsers "Generated Invoice"


    DoCmd.Close acForm, ("BookingInfo")


    '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 If


End Sub


Private Sub Command6_Click()


    DoCmd.Close , acForm("BookingInfo")

    'Closes the form


End Sub


Private Sub List1_Click()


End Sub
```

BookingInfoStaff

```
Option Compare Database

Private Sub Command11_Click()

    DoCmd.Close acForm, ("BookingInfoStaff")

End Sub

Public Sub Command5_Click()

    Dim db As DAO.Database
    Dim invoice As DAO.Recordset

    Set db = CurrentDb()
    Set invoice = db.OpenRecordset("Bookings", dbOpenDynaset)
    'Connect to bookings table

    If IsNull(Me.BookingIDBox.Value) Then
    MsgBox ("Please select a booking to view")
    Else
    bookid = Me.BookingIDBox.Value
    'Save BookingID number in box as bookid

    invoice.FindFirst ("BookingID = " & bookid)
    'Find first invoice where the BookingID matches bookid

    If invoice.NoMatch Then
        MsgBox ("No invoice with that Booking ID could be found")
        Exit Sub
    End If
    'If there's no match, show message box telling user.


    Dim RepName As String
    RepName = "Invoice - Booking #" & bookid & " RNDID" & Int((25000 - 10 + 1) * Rnd + 10)
    DoCmd.CopyObject , RepName, acReport, "Invoice Template"
    DoCmd.OpenReport RepName, acViewDesign, , , acHidden
    Reports(RepName).RecordSource = "SELECT ArtistName, BookingID, BookingStart, BookingEnd, BasePrice, BookingDate, Area, EquipmentLend, TotalCost FROM Bookings WHERE BookingID =" & Me.BookingIDBox.Value
    DoCmd.Save acReport, RepName
    DoCmd.OpenReport RepName, acViewPreview
    invoice.Close
    Globals.LoggingUsers "Generated Invoice"

    '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 BookingInfoStaff Form

    DoCmd.Close acForm, ("BookingInfoStaff")
    End If

End Sub

Private Sub Command6_Click()

    DoCmd.Close acForm, ("BookingInfoStaff")
    'Close Booking Info form
```

```
End Sub
```

CustomerForm

```
Option Compare Database

Private Sub Command11_Click()

   DoCmd.Close acForm, ("CustomerForm")
   'Close Customer Form

End Sub

Private Sub Command3_Click()

   DoCmd.OpenForm ("AddUser")
   'Open Add User Form


End Sub

Private Sub Command5_Click()

   DoCmd.OpenForm ("ModifyUser")
   DoCmd.Close acForm, ("CustomerForm")
   'Open Modify User Form
   'Close Customer Form

End Sub

Private Sub Command6_Click()

   DoCmd.Close acForm, ("CustomerForm")
   'Close Customer Form
```

```
End Sub
```

ModifyBooking

```
Option Compare Database

Private Sub Command11_Click()



 DoCmd.Close acForm, ("ModifyBooking")
 'Close Modify Booking form

End Sub

Public Sub Command5_Click()

 Dim db As DAO.Database
 Dim rs As DAO.Recordset

 Set db = CurrentDb()
 Set rs = db.OpenRecordset("Bookings", dbOpenDynaset)
 'Connect to database and Bookings table

 'Save ID of booking that user chooses

 If IsNull(Me.BookingIDBox.Value) Then
  MsgBox ("You must select a booking to modify.")
  'Produce an error if nothing is chosen.
  Else
  BookingMod = Me.BookingIDBox.Value
  rs.FindFirst ("BookingID = " & BookingMod)
  'Find the relevant data for the chosen booking in the table based on the booking ID

  DoCmd.Close acForm, ("ModifyBooking")
```

```
        DoCmd.OpenForm ("ModifyBooking_p2")
        'Close ModifyBooking, Open ModifyBooking_p2
    End If
End Sub

Private Sub Command6_Click()

    DoCmd.Close acForm, ("ModifyBooking")
    'Close ModifyBooking
    DoCmd.OpenForm ("AMain")
    'Open Artist Main Menu

End Sub
```

ModifyBooking_p2

```
Option Compare Database

Private Sub Command11_Click()

    DoCmd.Close acForm, ("ModifyBooking_p2")
    'Close ModifyBooking_p2 form

End Sub

Private Sub Command54_Click()

    Dim db As DAO.Database
    Dim table As DAO.Recordset

    Set db = CurrentDb()
    Set table = db.OpenRecordset("Bookings", dbOpenDynaset)
    'Connect to bookings table
```

```
   Dim oldDate As Date

   Dim newDate As Date

   Dim Hour As Integer

   'Declaring some variables as Date and Integer Variables. These will make calculations a bit simpler to understand.

   Hour = newDate

   oldDate = Me.BookingStart.Value

   newDate = DateDiff("h", oldDate, Me.BookingEnd.Value)

   'Due to changes with the code, this is more of a 'patch-up' job to keep the calculations working as intended.



   Dim varZ As Variant

   varZ = DLookup("[BookingStart]", _

   "Bookings", "CStr([Area]) = '" & CStr(Me.Area.Value) & "' And CStr([BookingDate]) = '" & CStr(Me.Date.Value) & "'")

   '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.


   Dim varA As Variant

   varA = DLookup("[BookingEnd]", _

   "Bookings", "CStr([Area]) = '" & CStr(Me.Area.Value) & "' And CStr([BookingDate]) = '" & CStr(Me.Date.Value) & "'")

   '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.


    If (varZ >= Me.BookingStart.Value And varZ <= newDate) Or (varA >= Me.BookingStart.Value And varZ <= newDate) Then

      MsgBox ("Sorry, that time or area is already booked. Please try another time or area.")

      'Checks 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.

   Else

   table.MoveFirst

   table.FindFirst ("BookingID = " & BookingMod)
```

```
    table.Edit
    table("BookingStart") = Me.BookingStart.Value
    table("BookingEnd") = Me.BookingEnd.Value
    table("EquipmentPrice") = Me.EquipmentPrice.Value
    table("EquipmentLend") = Me.Equip.Value
    table("BookingDate") = Me.Date.Value
    table("Area") = Me.Area.Value
    table("TotalCost") = Me.TotalPrice.Value
    table.Update
    'Replace data in relevant record with newly input data

    table.Close
    Set table = Nothing

    MsgBox ("Booking Successfully Modified")
    'Show messagebox confirming success

    Globals.LoggingUsers "Modified Booking"
    'Make entry into UserLog about modified booking

    DoCmd.Close acForm, ("ModifyBooking_p2")
    DoCmd.OpenForm ("AMain")

    End If

    'Close ModifyBooking_p2
    'Open Artist Main Menu

End Sub

Private Sub Command6_Click()

    DoCmd.Close acForm, ("ModifyBooking_p2")
    DoCmd.OpenForm ("ModifyBooking")
    'Close ModifyBooking_p2, Open ModifyBooking

End Sub

Public Sub Form_Open(Cancel As Integer)

    Dim db As DAO.Database
    Dim table As DAO.Recordset

    Set db = CurrentDb()
    Set table = db.OpenRecordset("Bookings", dbOpenDynaset)
    'Connect to booking table

    table.FindFirst ("BookingID = " & BookingMod)
    Me.BookingStart.Value = table.Fields("BookingStart")
    Me.BookingEnd.Value = table.Fields("BookingEnd")
    Me.Date.Value = table.Fields("BookingDate")
    Me.Area = table.Fields("Area")
    'Fill in the relevant data automatically based on what was selected in the previous form.

End Sub

Private Sub TotalPrice_Click()

    Dim db As DAO.Database
    Dim table As DAO.Recordset

    Set db = CurrentDb()
    Set table = db.OpenRecordset("Bookings", dbOpenDynaset)

    Dim Hrs As Integer
    Hrs = DateDiff("h", (BookingStart), (BookingEnd))
    If (Check33 = True) Then
        With Me
            .TotalPrice.Text = EquipmentPrice + 6 + (Hrs)
        End With
    Else
        With Me
            .TotalPrice.Text = EquipmentPrice + (3 * (Hrs)) + (Hrs)
        End With
    'Calculate and display price of booking based on equipment and user's discount
eligibility.

    End If
End Sub

Private Sub EquipmentPrice_Click()

    If (Equipment.ListIndex = 0) Then
        With Me
            .EquipmentPrice.Text = 0
        End With
    End If
    If (Equipment.ListIndex = 1) Then
        With Me
            .EquipmentPrice.Text = 2
        End With
    End If
    If (Equipment.ListIndex = 2) Then
        With Me
            .EquipmentPrice.Text = 4
        End With
    End If
    If (Equipment.ListIndex = 3) Then
        With Me
            .EquipmentPrice.Text = 6
        End With
    End If
    If (Equipment.ListIndex = 4) Then
        With Me
            .EquipmentPrice.Text = 8
        End With
    End If
    If (Equipment.ListIndex = 5) Then
        With Me
            .EquipmentPrice.Text = 10
        End With
    End If

'Calculate and display preset prices based on what list element is clicked
End Sub
```

ModifyUser

```
Option Compare Database
Dim FoundCustomer As Boolean

Private Sub Command11_Click()

    DoCmd.Close acForm, ("ModifyUser")
    'Close ModifyUser Form

End Sub

Private Sub Command6_Click()

    DoCmd.OpenForm ("CustomerForm")
    DoCmd.Close acForm, ("ModifyUser")
    'Open CustomerForm
    'Close Modify User Form

End Sub

Private Sub Detail_Click()

End Sub

Private Sub Form_Open(Cancel As Integer)
    FoundCustomer = False
    'Set FoundCustomer as False
End Sub


Private Sub Command32_Click()

    If IsNull(Me.Act.Value) Or IsNull(Me.ManT.Value) Or IsNull(Me.ManN.Value) Or IsNull(Me.Un.Value) Or IsNull(Me.pw.Value) Or IsNull(Me.Em.Value) Then
        MsgBox ("You must fill in the starred (*) fields.")
        Exit Sub
    End If

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

    Dim db As DAO.Database
    Dim table As DAO.Recordset

    Set db = CurrentDb()
    Set table = db.OpenRecordset("Artists", dbOpenDynaset)
    'Connect to artists table
```

```
If IsNull(Me.IDBox.Value) Then
   MsgBox ("Please enter a user ID")
   Exit Sub
End If
'Ask for user ID if not entered.

table.MoveFirst
table.FindFirst ("ArtistID = " & Me.IDBox.Value)
table.Edit
table("ActName") = Me.Act.Value
table("ManagerTitle") = Me.ManT.Value
table("ManagerName") = Me.ManN.Value
table("Username") = Me.Un.Value
table("Password") = Me.pw.Value
table("EmailAddress") = Me.Em.Value
table("Members") = Me.Mems.Value
table("U18") = Me.Check22.Value
table("CardNumber") = Me.Cardn.Value
table("CardCCV") = Me.CCV.Value
table("CardExpiry") = Me.Exp.Value
table.Update

'Update the record with changed information

table.Close
Set table = Nothing
'Close the Artists table to prevent corruption

MsgBox ("User Modified Successfully.")
'Confirm successful modificaton with messagebox

DoCmd.OpenForm ("SMain")
Globals.LoggingUsers "Modified User"
```

```
    'Open Staff Main Menu and add user modify to UserLog

    DoCmd.Close acForm, ("ModifyUser")
    'Close ModifyUser form.


End Sub

Private Sub IDFind_Click()


    Dim db As DAO.Database
    Dim table As DAO.Recordset

    Set db = CurrentDb()
    Set table = db.OpenRecordset("Artists", dbOpenDynaset)
    'Connect to Artists table

    If IsNull(Me.IDBox.Value) Or Not IsNumeric(Me.IDBox.Value) Then
        MsgBox ("Please enter a valid ID.")
        Exit Sub
    End If
    'If ID is invalid, ask for valid ID in messagebox

    table.FindFirst ("ArtistID = " & Me.IDBox.Value)
    'Find first instance of matching ID in the Artists table

    If table.NoMatch Then
        MsgBox ("No user with that ID was found.")
        Exit Sub
    End If
    'If none match, say so in messagebox.
    Me.Act.Value = table.Fields("ActName")
    Me.ManT.Value = table.Fields("ManagerTitle")
    Me.ManN.Value = table.Fields("ManagerName")
    Me.Un.Value = table.Fields("Username")
    Me.pw.Value = table.Fields("Password")
    Me.Em.Value = table.Fields("EmailAddress")
    Me.Mems.Value = table.Fields("Members")
    Me.Check22.Value = table.Fields("U18")
    Me.Cardn.Value = table.Fields("CardNumber")
    Me.CCV.Value = table.Fields("CardCCV")
    Me.Exp.Value = table.Fields("CardExpiry")
    'Fill in the form's fields with all relative info automatically.

    FoundCustomer = True
    'Set FoundCustomer to true

End Sub
```

Support

```
Option Compare Database

Private Sub Command11_Click()

    DoCmd.Close acForm, ("Support")
    'Close support form.

End Sub
```

UserLog

```
Option Compare Database

Private Sub Command11_Click()
```

```
    DoCmd.Close acForm, ("UserLog")
    'Close UserLog form

End Sub



Private Sub Command6_Click()

    DoCmd.Close acForm, ("UserLog")
    'Close UserLog form

End Sub
```

MODULES

GLOBALS

```
Public globUsername As String
Public globActname As String
Public BookingMod As Integer



Public Function SQLglobUsername() As String

  SQLglobUsername = globUsername
  'Saves SQLglobUsername as variable caled globUsername

End Function

Public Function getAct() As String

    Dim db As DAO.Database
    Dim rs As DAO.Recordset
    Dim qry As String
    Set db = CurrentDb()
    Set rs = db.OpenRecordset(qry)


    qry = "SELECT Artists.ActName FROM Artists WHERE (((Artists.Username)= & '" &
SQLglobUsername & "()));"


    rs.Close
    'Performs query to retrieve and save artist actname based on their globally saved
username


End Function


Public Function LoggingUsers(UserActivity As String)
    'Sets up global logging for UserLog, usable in any form.
    CurrentDb.Execute "INSERT INTO UserActivityLog (Username, UserActivity) Values ('"
& globUsername & "', '" & UserActivity & "')"
End Function
```

LIST OF SUBS & FUNCTIONS

Login

Private Sub Command11_Click()

This is the close button. Used to close form.

Public Sub LoginButton_Click()

Login button. Used to login to your account.

Private Sub RegisterButton_Click()

Register button. Used to view customer register form.

AMain

Private Sub Command11_Click()

This is the close button. Used to close form.

Private Sub Command2_Click()

Add Booking button. Opens Add Booking form.

Public Sub Command3_Click()

Modify Booking button. Opens Modify Booking form.

Private Sub Command5_Click()

Delete Account button. Prompts account deletion information box.

Private Sub Command6_Click()

Contact Support button. Opens Support form.

Private Sub Command7_Click()

Log Out button. Logs user out of the form.

Public Sub Command8_Click()

Show Invoice button. Opens Booking Info form.

Public Function Getuser()

Retrieves the user’s username, which is stored globally for future use.

AddBooking

Private Sub Close_Click()

This is the close button. Used to close form.

Public Sub CreateBtn_Click()

Create Booking button. Used to finalise booking creation. May produce error if form is inappropriately filled in. Also responsible for most calculations on this form.

Private Sub Command6_Click()

This is the back button. Used to return to Artist Control Panel.

Private Sub EquipmentPrice_Click()

Links list menu of equipment hire to text box. Used to display how much money is to be paid for equipment hire.

Private Sub TotalPrice_Click()

Calculates total price for the whole booking session. To be displayed in Total Price textbox.

AddStaff

Private Sub Command11_Click()

This is the close button. Used to close form.

Private Sub Command32_Click()

This is the Add Staff button. Used to finalise staff creation. May produce error if form is inappropriately filled in.

Private Sub Command6_Click()

This is the back button. Used to return to Staff Control Panel.

AddUser

Private Sub Command11_Click()

This is the close button. Used to close form.

Private Sub Command32_Click()

This is the Add User button. Used to finalise user creation. May produce error if form is inappropriately filled in.

Private Sub Command6_Click()

This is the back button. Used to return to Artist Control Panel.

BookingInfo

Private Sub Command11_Click()

This is the close button. Used to close form.

Public Sub Command5_Click()

This is the Generate Invoice button. Used to generate an invoice from the chosen booking.

Private Sub Command6_Click()

This is the back button. Used to return to artist control panel.

BookingInfoStaff

Private Sub Command11_Click()

This is the close button. Used to close form.

Public Sub Command5_Click()

This is the Generate Invoice button. Used to generate an invoice from the chosen booking.

Private Sub Command6_Click()

This is the back button. Used to return to staff control panel.

CustomerForm

Private Sub Command11_Click()

This is the close button. Used to close form.

Private Sub Command3_Click()

This is the add user button. Used to open AddUser form.

Private Sub Command5_Click()

This is the modify user button. Used to open ModifyUser form.

Private Sub Command6_Click()

This is the back button. Used to return to staff control panel.

ModifyBooking

Private Sub Command11_Click()

This is the close button. Used to close form.

Public Sub Command5_Click()

This is the modify button. Used to open ModifyBooking_p2 form if booking has been selected.

Private Sub Command6_Click()

This is the back button. Used to return to artist control panel.

ModifyBooking_p2

Private Sub Command11_Click()

This is the close button. Used to close form.

Private Sub Command54_Click()

This is the modify button. Used to apply edits to the selected booking if they are appropriate.

Private Sub Command6_Click()

This is the back button. Used to return to ModifyBooking Form.

Public Sub Form_Open(Cancel As Integer)

Fills in form with data of selected booking as soon as it opens.

Private Sub TotalPrice_Click()

Performs price calculations when total price box is clicked.

Private Sub EquipmentPrice_Click()

Performs price calculations when equipment price box is clicked.

ModifyUser

Private Sub Command6_Click()

This is the back button. Used to return to CustomerForm form.

Private Sub Form_Open(Cancel As Integer)

Sets a variable as soon as form opens.

Private Sub Command32_Click()

Modify button. Applies all edits to user, if they are appropriate.

Private Sub IDFind_Click()

Automatically fills in all relevant info based on which user was selected from the list.

SMain

Private Sub Command11_Click()

This is the close button. Used to close form.

Private Sub Command2_Click()

This is the view customers button. Used to open CustomerForm form.

Private Sub Command5_Click()

Delete Account button. Prompts account deletion information box.

Private Sub Command6_Click()

This is the add staff button. Used to open AddStaff form.

Private Sub Command7_Click()

Log Out button. Logs user out of the form.

Public Sub Command8_Click()

Artist Panel Debug Button. Used to view artist panel.

Public Function Getuser()

Retrieves current username.

Private Sub Command9_Click()

User log button. Used to open UserLog form.

Support

Private Sub Command11_Click()

This is the close button. Used to close form.

UserLog

Private Sub Command11_Click()

This is the close button. Used to close form.

Private Sub Command6_Click()

This is the back button. Used to return to staff control panel

Maintenance Documentation.

List of Variables

MODULE:GLOBALS

Dim qry As String: Variable containing SQL query which finds artist name based on their username.

Public globUsername As String: Saves username of logged in user as a global variable

Public globActname As String: Saves artist's actname as a global variable for later use.

Public BookingMod As Integer: Sets BookingMod as a global variable for later use.

RECORD:ARTISTS

ArtistID As Integer: Autonumber which allocates an ID to artist accounts.

ActName As String: The name of the user's act.

ManagerName As String: The name of the account's user / manager.

Username As String: The unique account username used to login and keep track of specific information.

Password As String: The account's password used to login to the system.

EmailAddress As String: The email address associated with the user's account.

U18 As Boolean: A checkbox confirming whether the user is under 18 or not.

CardNumber As String: The debit card number associated with the user's account.

CardCCV As String: The debit card security code associated with the user's account.

CardExpiry As String: The debit card expiry date associated with the user's account.

ManagerTitle As String: The title of the account's user / manager.

Members As Integer: The number of members in the user's act.

RECORD:STAFF

StaffID As Integer: Autonumber which allocates an ID to staff accounts.

FullName As String: The full name of the staff member.

Title As String: The title of the staff member.

EmailAddress As String: The email address of the staff member.

DateOfBirth As Date: The date of birth of the staff member.

Username As String: The username of the staff member used to login and keep track of specific information.

Password As String: The password of the staff member used to log into the system.

RECORD:BOOKINGS

BookingID As Integer: Autonumber which allocates an ID to individual bookings.

BookingStart As Date: The time that the booking starts.

BookingEnd As Date: The time that the booking ends.

BookingDate As Date: The date that the booking is scheduled for.

ArtistName As String: The name of the artist that scheduled the booking.

TotalCost As Real: The total cost of the booking.

EquipmentPrice As Real: The total price of renting equipment for the booking.

BasePrice As Real: The type of booking made which determines the base price for calculations.

U18 As Boolean: A checkbox confirming whether the user is under 18 or not.

EquipmentLend As String: A note of all of the equipment that the user wishes to borrow for their booking session.

Area As String: The area that the booking will take place in.

ArtistID As Integer: The ID of the artist account that has created the booking.

RECORD:USERLOG

ID As String: The unique identifier for the log entry.

Timestamp As Date: A timestamp to identify when the action in the userlog happened.

Username As String: The username of the person attached to the action in the userlog.

Useractivity As String: A short description of the action logged.

List of Objects

AddBooking.

Command Button: Command6; The back button used to return to the previous form.

Command Button: Close; The close button used to close the current form.

Command Button: CreateBtn; The 'Create Booking' button used to finalise booking creation.

Textbox: EquipmentPrice; Used to show calculation of equipment borrowing price.

Textbox: Equip; User can input what equipment they wish to borrow.

Textbox: Date; User can select a date for their booking.

Textbox: TotalPrice; Used to show calculation of total booking price.

Textbox: CardNum: User can input their card number for payment.

Textbox: CCV: User can input their card security code for payment.

Checkbox: Check33; User can select whether they are under 18.

Combo Box: BookingStart; The box used to select a booking start time.

Combo Box: BookingEnd; The box used to select a booking end time.

Combo Box: Area; The box used to select a booking area.

List Box: Equipment; A list box with selectable equipment options.

AddStaff

Command Button: Command6; The back button used to return to the previous form.

Command Button: Command11; The close button used to close the current form.

Command Button: Command32; Used to finalise creation of staff member account.

Combo Box: ManT; Used to choose title of staff member.

Textbox: Act; Used to enter full name of staff member.

Textbox: Un; User can input their desired username.

Textbox: pw; User can input their desired password.

Textbox: Pw2; User can verify their desired password.

Textbox: Em; User can input their email address.

Textbox: Mems; User can input their date of birth.

AddUser

Command Button: Command6; The back button used to return to the previous form.

Command Button: Command11; The close button used to close the current form.

Command Button: Command32; Used to finalise creation of staff member account.

Combo Box: ManT; Used to choose account manager's title.

Check Box: Check22; User can choose whether they're under 18.

Textbox: Act; User can input their act name.

Textbox: ManN; Used to input account manager's name.

Textbox: Un; User can input their desired username.

Textbox: pw; User can input their desired password.

Textbox: Pw2; User can verify their desired password.

Textbox: Em; User can input their email address.

Textbox: Mems; User can input the number of members in their act.

Textbox: CardN: User can input their card number for payment.

Textbox: CCV: User can input their card security code for payment.

Textbox: Exp; User can input their card expiry date for payment.

AMain

Command Button: Command11; The close button used to close the current form.

Command Button: Command2; Used to open create booking form.

Command Button: Command3; Used to open modify booking form.

Command Button: Command5; Used to open delete account prompt.

Command Button: Command6; Used to open support form.

Command Button: Command7; Used to log out of the system.

Command Button: Command8; Used to open the bookinginfo form to get an invoice.

BookingInfo

Command Button: Command6; The back button used to return to the previous form.

Command Button: Command11; The close button used to close the current form.

Command Button: Command5; Generate invoice button used to generate an invoice.

List Box: List1; Used to view and select a booking on user's account.

Text Box: BookingIDBox; Used to view ID of booking that is chosen.

BookingInfoStaff

Command Button: Command6; The back button used to return to the previous form.

Command Button: Command11; The close button used to close the current form.

Command Button: Command5; Generate invoice button used to generate an invoice.

List Box: List1; Used to view and select a booking across all accounts.

Text Box: BookingIDBox; Used to view ID of booking that is chosen.

<u>CustomerForm</u>

Command Button: Command6; The back button used to return to the previous form.

Command Button: Command11; The close button used to close the current form.

Command Button: Command3; Used to open AddUser form.

Command Button: Command5; Used to open ModifyUser form.

List Box: CustomerList; View all customers (artists) on the database.

<u>Login</u>

Picture: Image10; The company logo.

Command Button: Command11; The close button used to close the current form.

Command Button: LoginButton; The button used to login to the system.

Command Button: RegisterButton; Takes user to the AddUser form.

Textbox: UsernameBox; User inputs their username here.

Textbox: PasswordBox; User inputs their password here.

<u>ModifyBooking</u>

Command Button: Command6; The back button used to return to the previous form.

Command Button: Command11; The close button used to close the current form.

Command Button: Command5; Modify button which loads chosen booking into ModifyBooking_p2 form.

List Box: List1; Used to view and select a booking across all accounts.

Text Box: BookingIDBox; Used to view ID of booking that is chosen.

<u>ModifyBooking_p2</u>

Command Button: Command6; The back button used to return to the previous form.

Command Button: Close; The close button used to close the current form.

Command Button: Command54; The ‘Modify Booking’ button used to finalise modifications.

Textbox: EquipmentPrice; Used to show calculation of equipment borrowing price.

Textbox: Equip; User can input what equipment they wish to borrow.

Textbox: Date; User can select a date for their booking.

Textbox: TotalPrice; Used to show calculation of total booking price.

Checkbox: Check33; User can select whether they are under 18.

Combo Box: BookingStart; The box used to select a booking start time.

Combo Box: BookingEnd; The box used to select a booking end time.

Combo Box: Area; The box used to select a booking area.

List Box: Equipment; A list box with selectable equipment options.

ModifyUser

Command Button: Command6; The back button used to return to the previous form.

Command Button: Command11; The close button used to close the current form.

Command Button: Command32; Used to finalise creation of staff member account.

CommandButton: IDFind; Used to populate other textboxes based on selected user.

Combo Box: ManT; Used to choose account manager's title.

Check Box: Check22; User can choose whether they're under 18.

List Box: List11; Displays all artists on the system.

Textbox: IDBox; Displays ID of user selected from list.

Textbox: Act; User can input their act name.

Textbox: ManN; Used to input account manager's name.

Textbox: Un; User can input their desired username.

Textbox: pw; User can input their desired password.

Textbox: Pw2; User can verify their desired password.

Textbox: Em; User can input their email address.

Textbox: Mems; User can input the number of members in their act.

Textbox: CardN: User can input their card number for payment.

Textbox: CCV: User can input their card security code for payment.

Textbox: Exp; User can input their card expiry date for payment.

SMain

Command Button: Command11; The close button used to close the current form.

Command Button: Command2; Used to open CustomerForm form.

Command Button: Command3; Used to open BookingInfoStaff form.

Command Button: Command5; Used to open delete account prompt.

Command Button: Command6; Used to open AddStaff form.

Command Button: Command7; Used to log out of the system.

Command Button: Command8; Used to open AMain form.

Command Button: Command9; Used to open UserLog form.

Support

Command Button: Command11; The close button used to close the current form.

UserLog

Command Button: Command11; The close button used to close the current form.

Command Button: Command6; The back button used to return to the previous form.

List Box: List1; Displays the userlog.

InvoiceTemplate

Picture: Image20; Logo of the company.

Textbox: Text3; Displays booking ID.

Textbox; Text7; Displays price category.

Textbox: Text11; Displays booking duration.

Textbox: Text13; Displays booking date.

Textbox: Text15; Displays booking area.

Textbox: Text17; Displays equipment borrowed.

Textbox: Text23; Shows total price.

Check Box: Check21; Allows user to tick off if they have paid.