

CIS 208 Course Project

James Aniciete

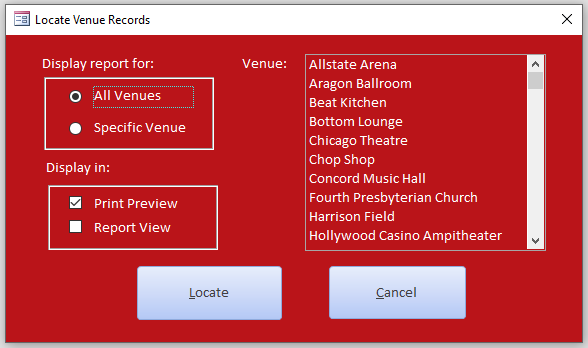
December 3, 2019

**Part 1: Business Case**

For my database, the user will be required to make three selections on a form. The first selection will be between two buttons to determine whether or not the report will be filtered by a specific venue. Then, the second selection will be between two check boxes to determine in which view the report will be displayed. Lastly, the third selection will be done by selecting a venue name from a list. Based on these selections, a report will be displayed with the following information: Concert ID, Headliner, Concert Date, Venue, Ticket Description, Amount, Face Price, Fees, Taxes, Resale Surcharge, and Ticket Price. Here, the Ticket Price will be calculated as the product of the Amount and the sum of the Face Price, Fees, Taxes, and Resale Surcharge.

**Part 2: Pseudocode**

Form (frmLocateVenueRecords):



Module (mdlLocateVenueRecords):

1. Open the form named frmLocateVenueRecords.

Button (cmdLocate):

1. Use the IsNull function to confirm if a selection was made from the list box named lstVenue. If no selection was made, display a MsgBox telling the user to make a selection and exit the sub.
2. Option button selected in the option group control named orgReportType:

= 1 Display the report showing all of the venues’ records.

* 1. Check box selected in the option group control named orgDisplayView:

= 1 Display the report in Print Preview.

= 2 Display the report in Report View.

= 2 Assign the selected list box item to a string variable named strVenue.

Display the report showing only the information for the venue stored in the

strVenue variable.

1. Check box selected in the option group control named orgDisplayView:

= 1 Display the report in Print Preview.

= 2 Display the report in Report View.

Else

Use the MsgBox statement to display a message indicating that the report preview

was cancelled.

Button (cmdCancel):

1. Close the form named frmLocateVenueRecords.

Module (mdlCalcTicketPrice):

1. Using DAO object variables, create a new field named “VBA Ticket Price” with a currency data type in the table named tblTicketPrices and append it to the table’s definition.
2. Open a recordset that includes all of the records stored in the tblTicketPrices table.
3. Repeat the following for each record in the recordset:
   1. Assign the value of the “VBA Ticket Price” field to be the expression: [Amount] \* ([FacePrice] + [Fees] + [Taxes] + [ResaleSurcharge]).
   2. Save the change to the record.
   3. Move the record pointer to the next record.
4. Close the recordset.
5. Dissasociate all object variables used for the calculation.
6. Display a message box telling the user if the ticket prices were successfully calculated.

**Part 3: VBA Coding**

mdlLocateVenueRecords Module:

Option Compare Database

Option Explicit

Public Function LocateVenueRecords()

'display custom dialog box

DoCmd.OpenForm FormName:="frmLocateVenueRecords"

End Function

cmdLocate Button:

Option Explicit

Option Compare Database

Private Sub cmdLocate\_Click()

'declare variable

Dim strVenue As String

'if no selection was made from the list box, tell the user with a MsgBox & \_

exit the sub

If IsNull(Me!lstVenue) Then

MsgBox Prompt:="Please make a selection from the list.", Buttons:=vbOKOnly + \_

vbCritical, Title:="No Selection Made"

Exit Sub

End If

'based on option button & check mark selections, display the report with \_

the corresponding WhereCondition and View

Select Case orgReportType.Value

Case 1 'display records for all venues

If orgDisplayView.Value = 1 Then 'display in Print Preview

DoCmd.OpenReport ReportName:="rptConcertOrders", View:=acViewPreview

ElseIf orgDisplayView.Value = 2 Then 'display in Report View

DoCmd.OpenReport ReportName:="rptConcertOrders", View:=acViewReport

End If

Case 2 'display records for a specific venue

'assign the selected venue name from the list box to the string variable

strVenue = lstVenue.Column(0)

If orgDisplayView.Value = 1 Then 'display in Print Preview

DoCmd.OpenReport ReportName:="rptConcertOrders", View:=acViewPreview, \_

WhereCondition:="Venue = '" & strVenue & "'"

ElseIf orgDisplayView.Value = 2 Then 'display in Report View

DoCmd.OpenReport ReportName:="rptConcertOrders", View:=acViewReport, \_

WhereCondition:="Venue = '" & strVenue & "'"

End If

Case Else

MsgBox Prompt:="Report preview cancelled.", Buttons:=vbOKOnly + vbCritical, \_

Title:="Error"

End Select

End Sub

cmdCancel Button:

Private Sub cmdCancel\_Click()

DoCmd.Close

End Sub

mdlCalcTicketPrice Module:

Option Compare Database

Option Explicit

Public Function CalcTicketPrice()

'declare variables

Dim dbsData As DAO.Database, tdfTicketPrices As DAO.TableDef, \_

fldTicketPrice As DAO.Field, rstTicketPrices As DAO.Recordset

'assign addresses to object variables

Set dbsData = CurrentDb

Set tdfTicketPrices = dbsData.TableDefs("tblTicketPrices")

'create a currency field for the calculation

Set fldTicketPrice = tdfTicketPrices.CreateField("VBA Ticket Price", \_

Type:=dbCurrency)

'append that field to tblTicketPrices's table definition

tdfTicketPrices.Fields.Append fldTicketPrice

'assign address to recordset variable

Set rstTicketPrices = dbsData.OpenRecordset("tblTicketPrices", dbOpenDynaset)

'processing/calculation

With rstTicketPrices

'move to the first record of the recordset

.MoveFirst

Do Until .EOF = True

'prepare recordset for editing

.Edit

'calculate the ticket price

.Fields("VBA Ticket Price").Value = .Fields("Amount") \* \_

(.Fields("FacePrice").Value + .Fields("Fees").Value + \_

.Fields("Taxes").Value + .Fields("ResaleSurcharge").Value)

'save change to current record

.Update

'move to the next record of the recordset

.MoveNext

Loop

'close the recordset

.Close

End With

'disassociate the object variables from their respective objects

Set dbsData = Nothing

Set tdfTicketPrices = Nothing

Set fldTicketPrice = Nothing

Set rstTicketPrices = Nothing

'display message if successful for macro

MsgBox Prompt:="Ticket prices have been calculated.", \_

Buttons:=vbOKOnly, Title:="Successful Calculations"

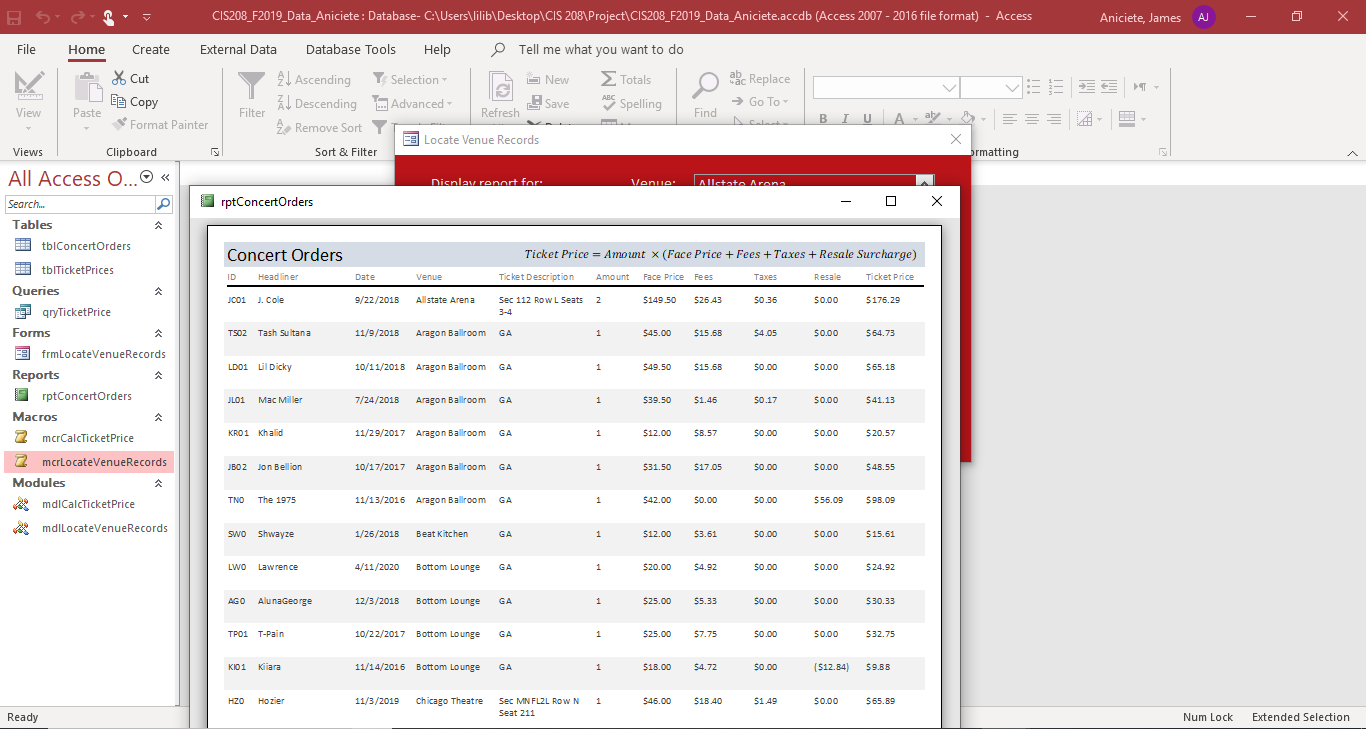
End Function

**Part 4: Testing**

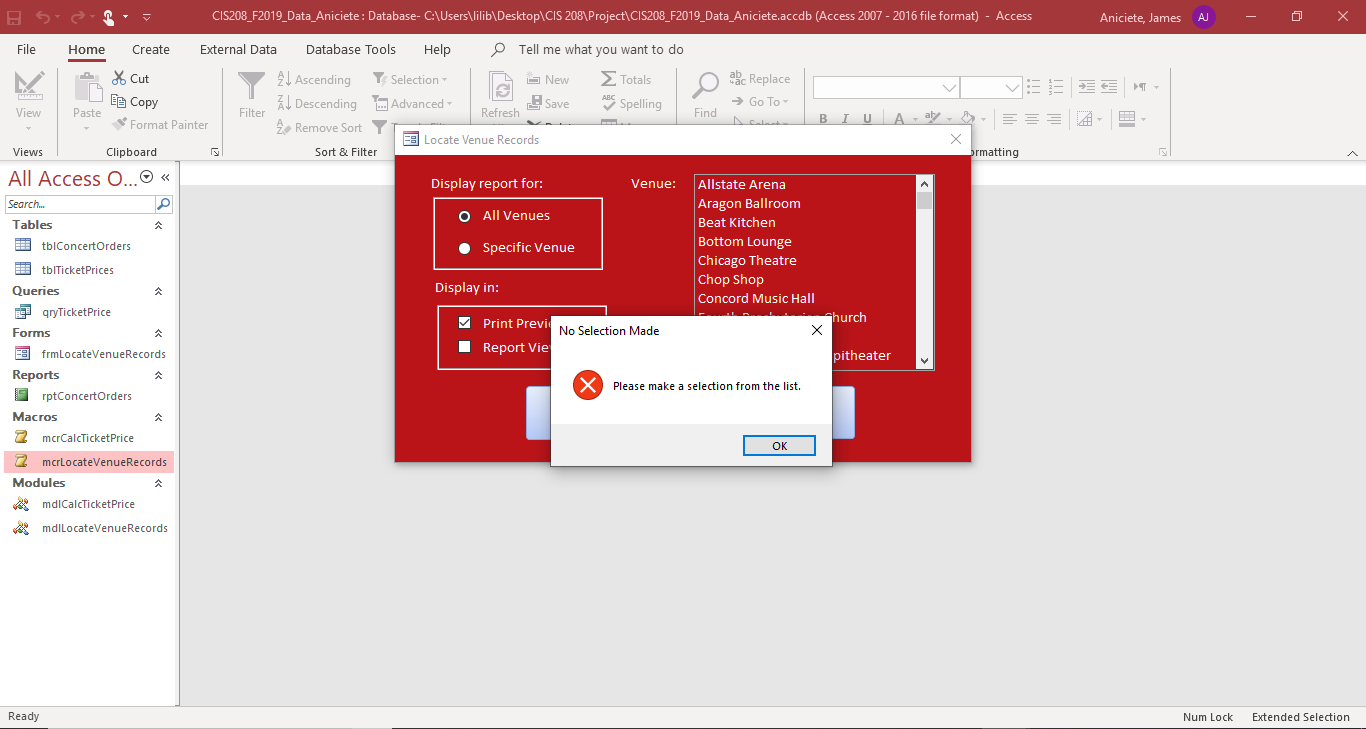
Summary Table for frmLocateVenueRecords Form:

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | (Filter, View, Venue) | Expected Value | Actual Value Equals Expected? |
| 1 | (All, PP, Chop Shop) | Report displayed in Print Preview with all of the venues’ records. | Yes |
| 2 | (All, PP, Null) | MsgBox displays telling the user to make a selection from the list box. | Yes |
| 3 | (All, RV, Chop Shop) | Report displayed in Report View with all of the venues’ records. | Yes |
| 4 | (All, RV, Null) | MsgBox displays telling the user to make a selection from the list box. | Yes |
| 5 | (Specific, PP, Chop Shop) | Report displayed in Print Preview with only the records for the Chop Shop venue. | Yes |
| 6 | (Specific, PP, Null) | MsgBox displays telling the user to make a selection from the list box. | Yes |
| 7 | (Specific, RV, Chop Shop) | Report displayed in Report View with only the records for the Chop Shop venue. | Yes |
| 8 | (Specific, RV, Null) | MsgBox displays telling the user to make a selection from the list box. | Yes |

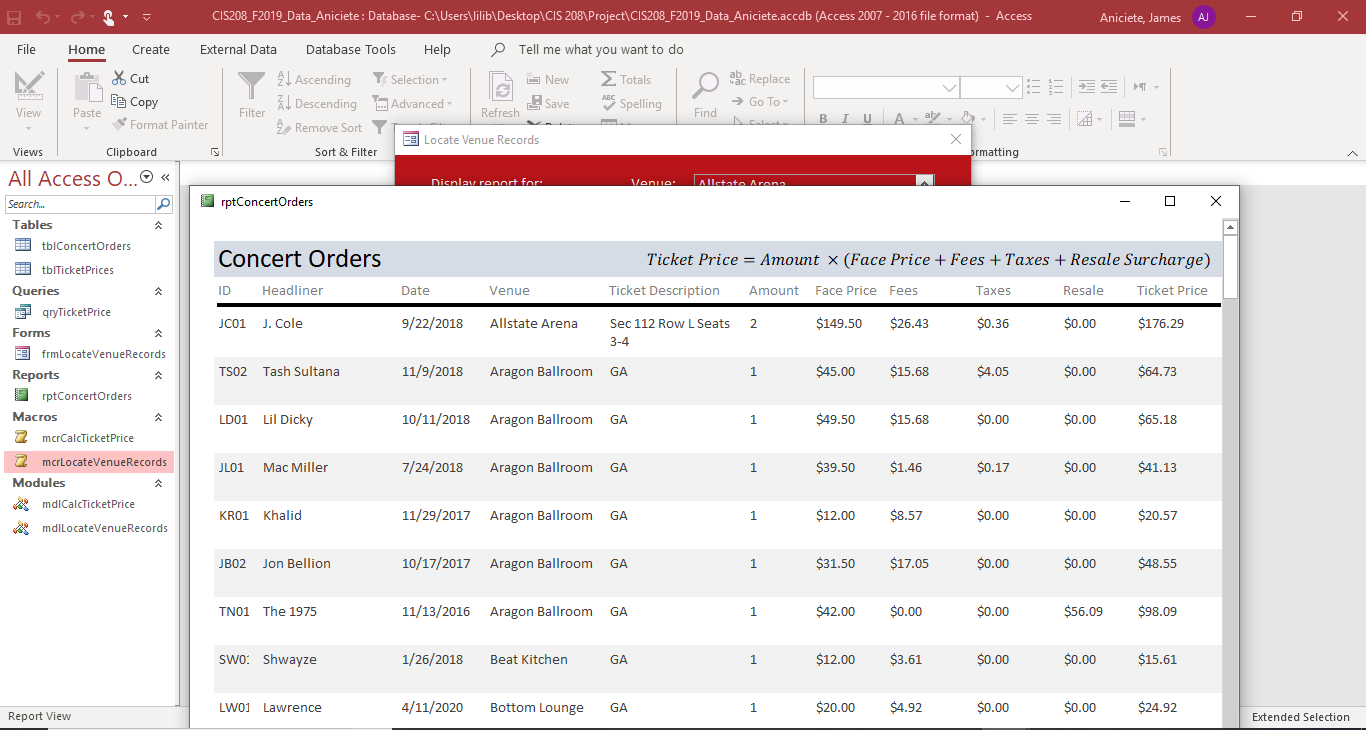
Test Case 1 (All Venues, Print Preview, Chop Shop):



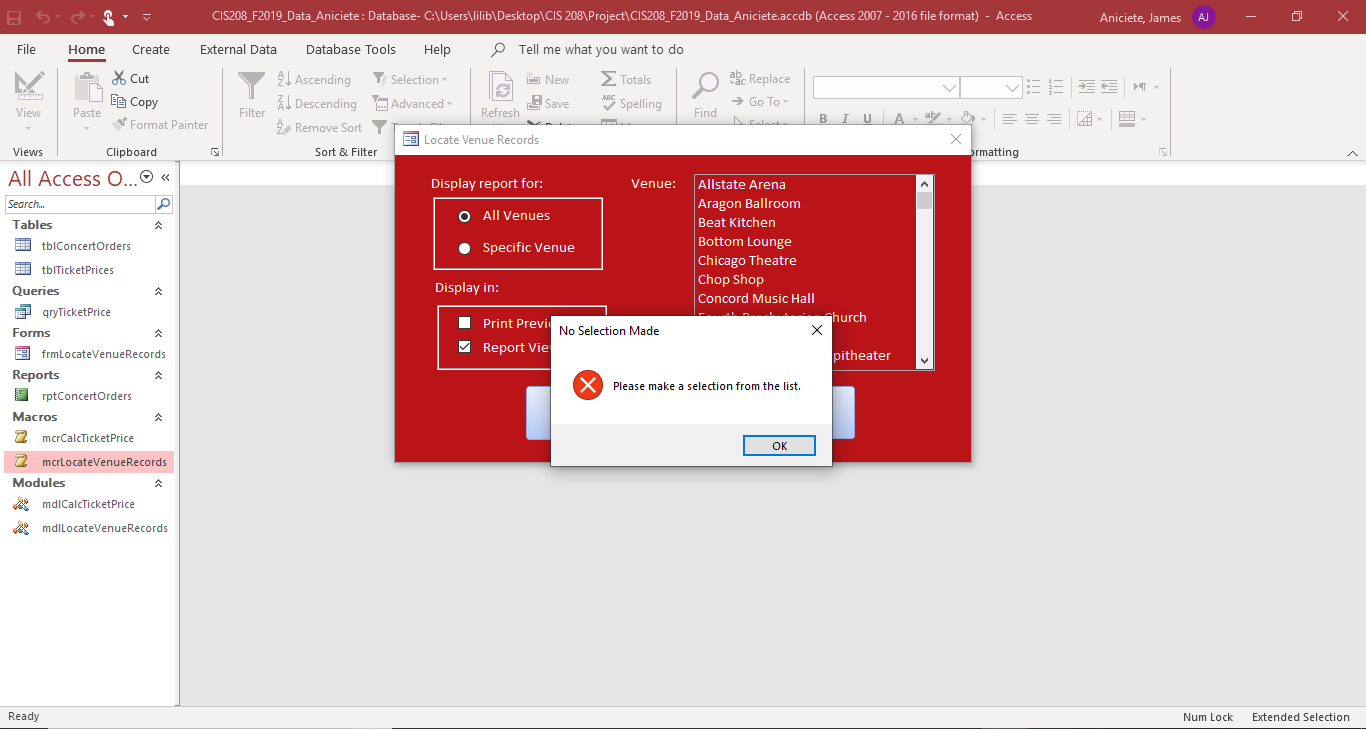
Test Case 2 (All Venues, Print Preview, Null):



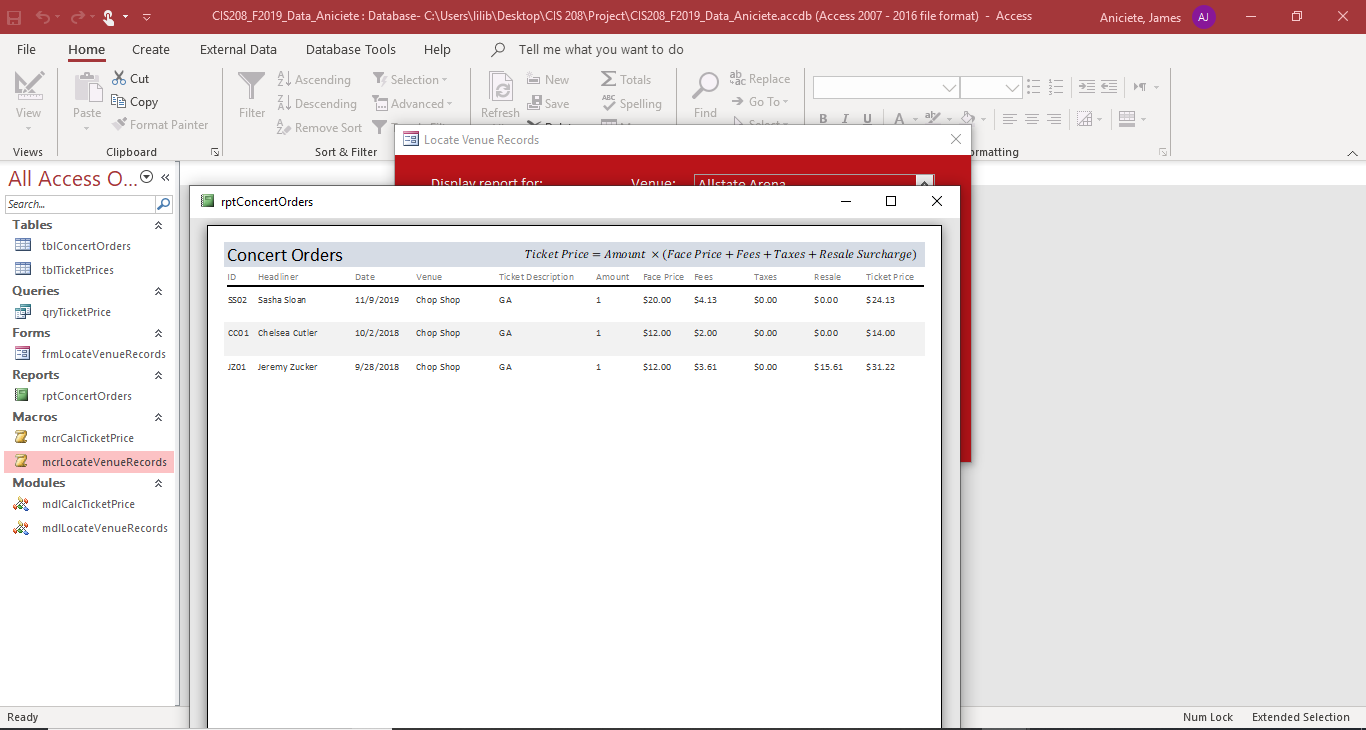
Test Case 3 (All Venues, Report View, Chop Shop):



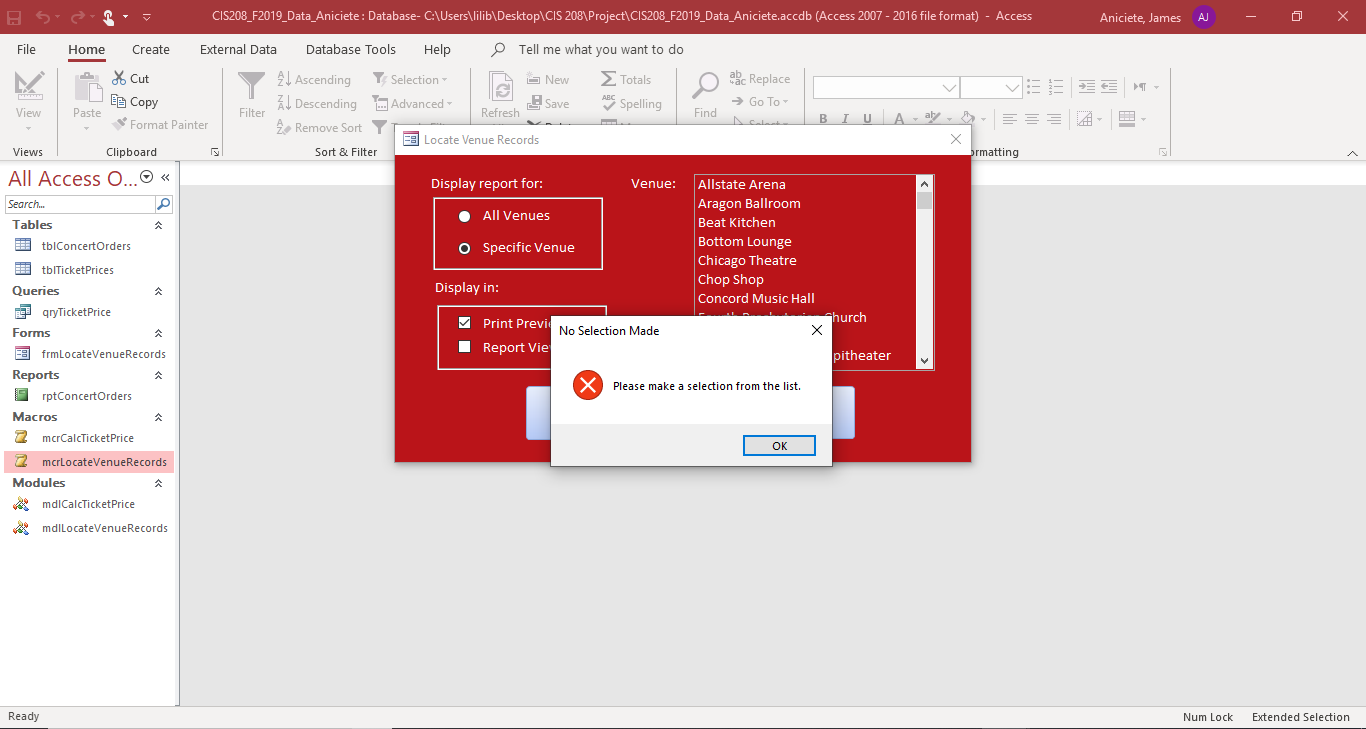
Test Case 4 (All Venues, Report View, Null):



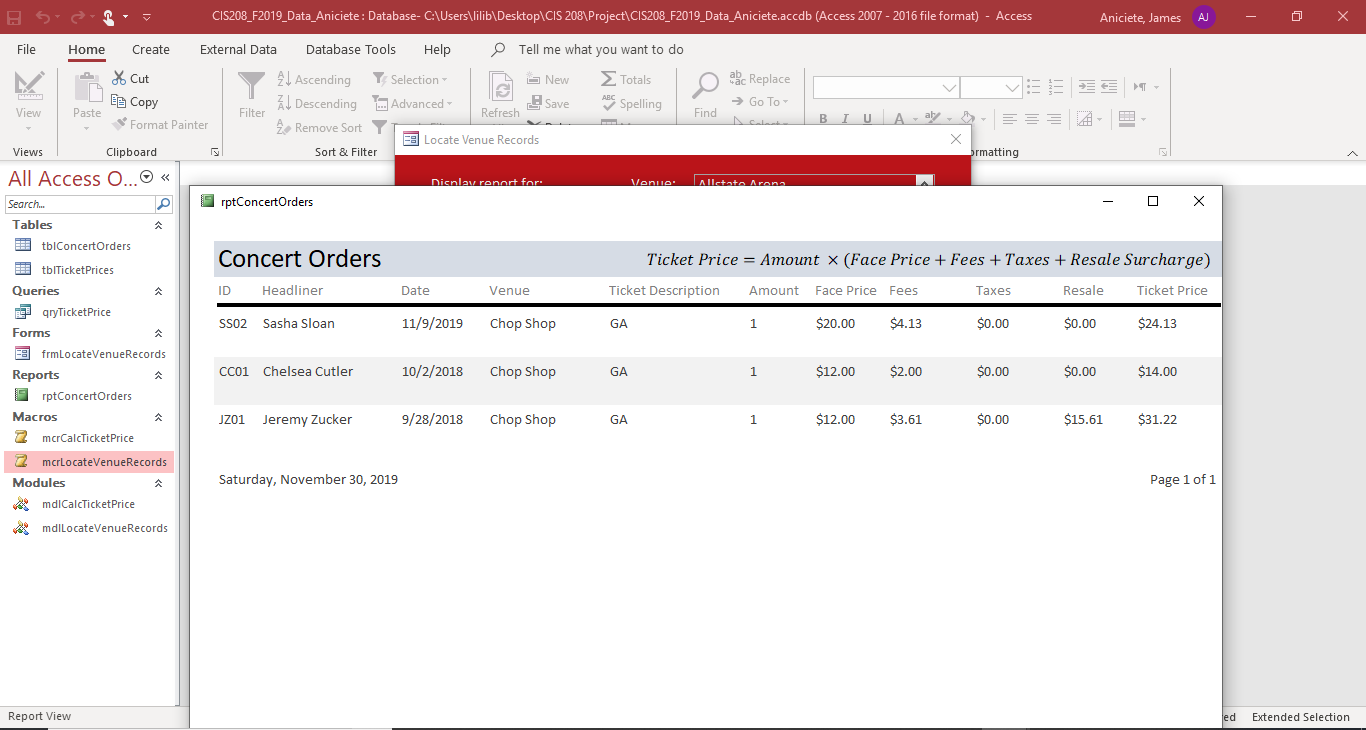
Test Case 5 (Specific Venue, Print Preview, Chop Shop):



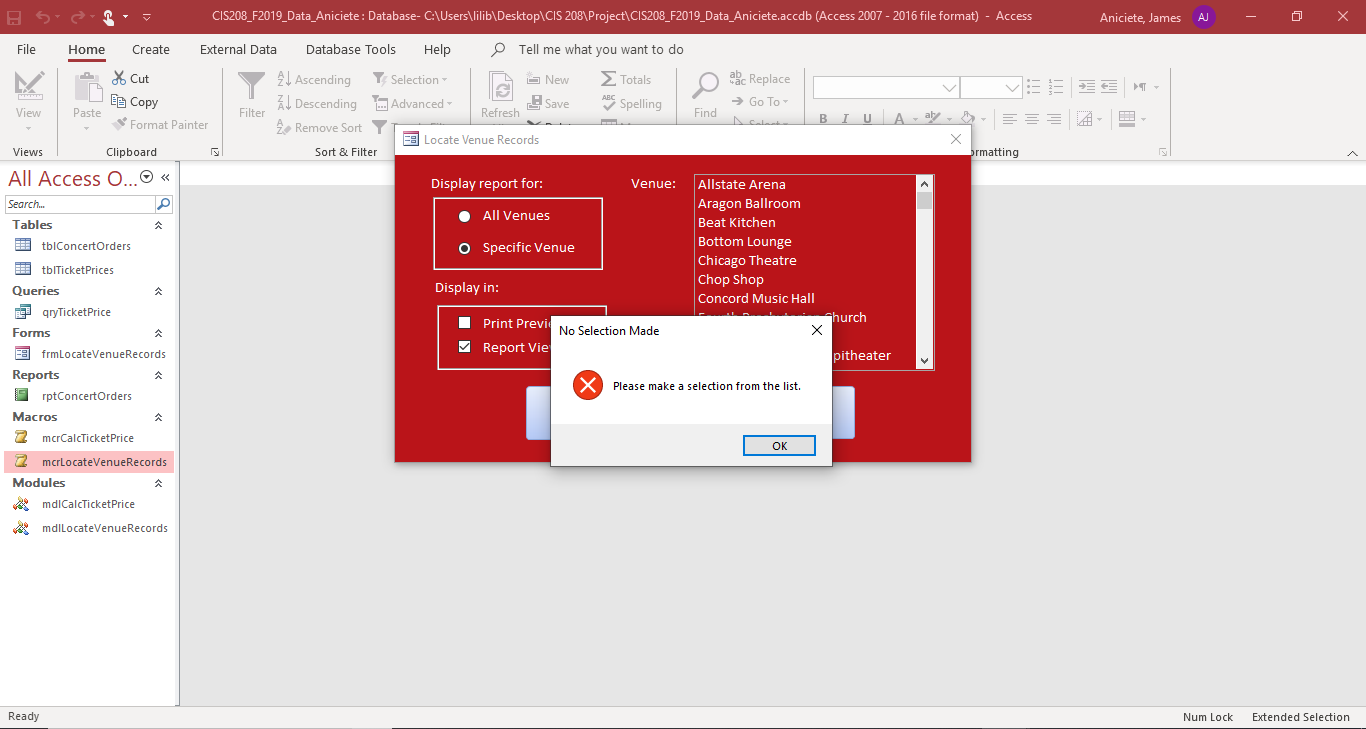
Test Case 6 (Specific Venue, Print Preview, Null):



Test Case 7 (Specific Venue, Report View, Chop Shop):



Test Case 8 (Specific Venue, Report View, Null):



mdlCalcTicketPrice:

For this module, there were two tests. The first test checked if the message box displayed, indicating that the corresponding macro executed successfully. The expected output is a message box entitled “Successful Calculations” with a prompt saying that ticket prices have been calculated and an OK button. The result of this test was that the expected message box was displayed. The second test checked if the VBA Ticket Prices were equal to the Access Ticket Prices. The VBA-calculated Ticket Prices were expected to have the same values as those that were calculated with an equivalent Access expression. The actual values of the VBA Ticket Prices did in fact equal the Access Ticket Prices.

