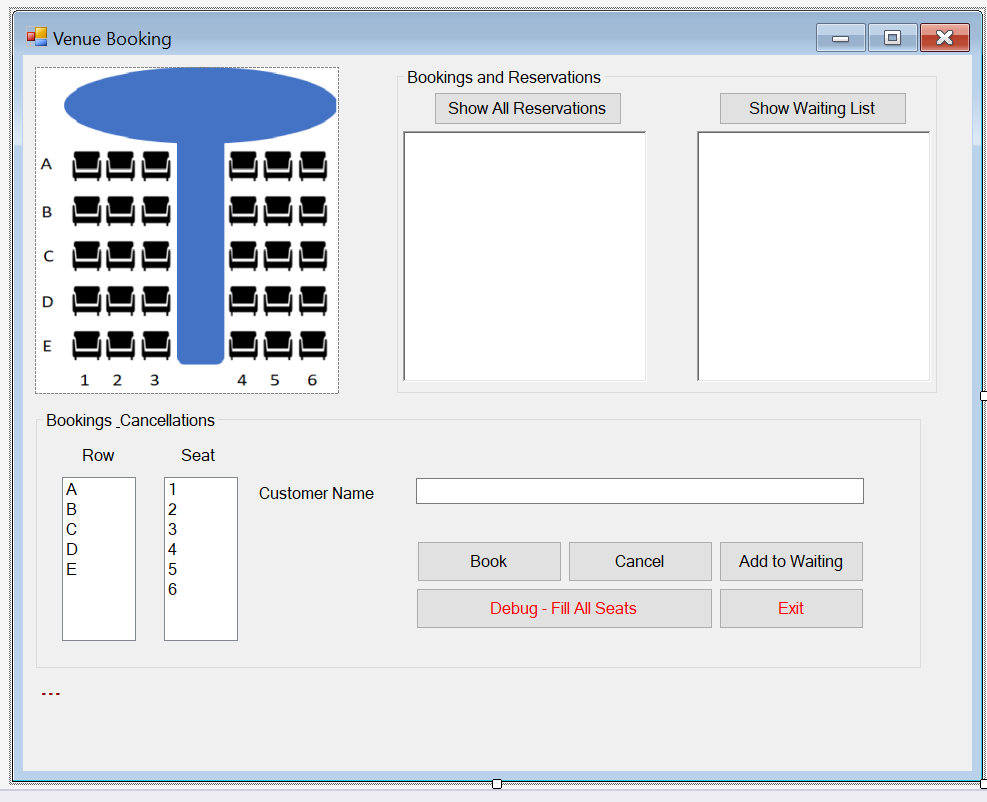
## Programming Concepts II – Assignment 1 – Venue Booking

You are working for a concert/fashion/convention venue, with a stage, a runway and seating along the runway. You are writing a booking system, storing bookings in a 2-dimensional array.



List Boxes … use their .SelectedIndex to index into 2-dimension booking array in code

Draw an picture or a pattern of buttons … not for storage

Rich Text Boxes

… not List Boxes

… not for storage

Label for messages

1. ~~The booking clerk should be able to click “Show All Reservations” to display a list of all seats in the venue, one line per seat~~. ~~It should show the row letter and seat number for each seat, along with the customer’s name, if the seat is booked. The listing should be one line per seat, like this:~~

~~A1 –~~

~~A2 –~~

~~A3 – Russell Crowe~~

~~A4 – Shania Twain~~

* 1. Hint: you can use the array’s row index to index into the ListBox, when translating it into a letter.

1. ~~The clerk would select a seat using the row/seat list boxes and click “Book”, adding the customer’s name to the booking array.~~
   1. ~~If the selected seat is already booked, a message should display to that effect, along with a list of the seats that are currently available. If no seats are available, it should say so.~~
   2. ~~If all seats are booked, the clerk can add the customer to the waiting list. It has a maximum of 10 people. If the waiting list is full, show a message to that effect~~. ~~“Show Waiting List” should show the customers already waiting, in first-to-last order.~~
   3. ~~If seats are available, the “Add to Waiting” button will just show the message “Seats are available”, with a list of the seats that are not booked.~~
   4. ~~Clicking the “Status” button will show the status of the selected seat~~.
2. Cancelling a customer’s booking:
   1. ~~Select the row & seat and click “Cancel”. A message box should pop up to show the customer’s name and ask for confirmation.~~
   2. ~~If the cancellation works, and the waiting list is not empty, remove the first customer from the list and give her/him the newly vacated seat.~~
3. ~~You are to use a~~ **~~multi-dimensional array to record seat bookings~~** ~~(ie: person’s name at row/column intersection), and a~~ **~~single-dimensional array for the waiting list~~**~~.~~
   1. **~~You are not to store bookings or the waiting list in any of the form controls~~**
   2. ~~Use a loop to format information from the arrays when “Show All Reservations” or “Show Waiting List” are clicked.~~ ~~Display this in the RichTextBoxes, one line per seat or waiting customer (RichTextBoxes honour “\n”, which TextBoxes don’t).~~
4. Show a message when
   1. ~~“Book” is pressed but either the customer’s name, row and/or seat is not selected.~~
   2. ~~“Book” is pressed when the chosen seat is already taken~~
   3. ~~“Add to Waiting List” is pressed when there are seats available.~~
   4. ~~“Cancel” is pressed without specifying the row and seat~~.
   5. ~~“Cancel” is pressed but the seat is not booked~~
   6. ~~After a successful booking, including one from the waiting list~~
   7. ~~After successfully cancelling a seat.~~
   8. ~~After successfully adding to the waiting list.~~
5. ~~Some of the above actions can generate two messages … “seat cancelled” and “seat booked from waiting list” or “name is required”, “please select a row” and “please select a seat” All relevant messages should show at the same time, on separate lines (labels honour “\n”).~~
6. ~~Add a “Fill All” button. Clicking this button will fill all 30 seats with the same customer name.~~
7. ~~Don’t restrict the length of the customer’s name. For example, I should be able to book a person with name “C”.~~
8. Remember, for all programs:
   1. ~~Program comments before the class, describing what it does & who wrote it.~~
   2. ~~Comments before each method describing what they do … simply repeating the method name Is not acceptable~~.
   3. ~~Form controls and program variable names should be camel notation, full meaningful words~~
   4. ~~Form controls can have a prefix denoting their control type: txt, btn, lbl, lst, etc.~~
   5. ~~Variables can not have datatype prefixes or suffixes (ie: no Hungarian notation: strName~~)
   6. ~~Methods you write should be Pascal notation. Generated event handlers are named after their control, so they can be camel notation.~~

## Hand In

1. Zip and upload your solution folder to the assignment drop-box on eConestoga.ca.