**Seat/Client Assignment Second Pass (Final Pass)**

**Overview**

The purpose of the final pass is to create an application that assign clients to a 10x10 matrix similar to booking the clients for a movie theatre. The GUI for this project is implemented using Java Swing. Clients are loaded from the csv file to a JList. Users can manually book and unbook the clients.

**Core Features**

* JList containing a list of clients.
* JToggle buttons allows booking and unbooking for each seat.
* JButtons allows resetting the seating matrix.
* JButtons shows the current seat assignment in the console.
* Assign clients to seats manually:
* Can book and unbook any selected client to a seat in the matrix.
* Display clients currently assigned to the seating matrix to console.

**Constraints**

* Limited amount of time for the delivery of the Project.
* Limited GUI options to use to develop the front end.

**Background**

The core purpose of this app is to allow user to manually book a list of clients to a matrix where all the seats are vacant. The initial GUI set up for the app is shown in **Figure.1**.

Several classes are used directly from the first pass with slight modifications.

* The Client Class
* The ClientStore Class
* The Seat Class

The Client class contains the first name, last name, and ID of the client. The only difference from the first pass is that now the client class contains an overwritten toString method to display the full name each client object. The ClientStore class loads the clients from the csv file. The Seat Class is a replica of the Client Class with additional data field row and column to store the position of the seat in the matrix.



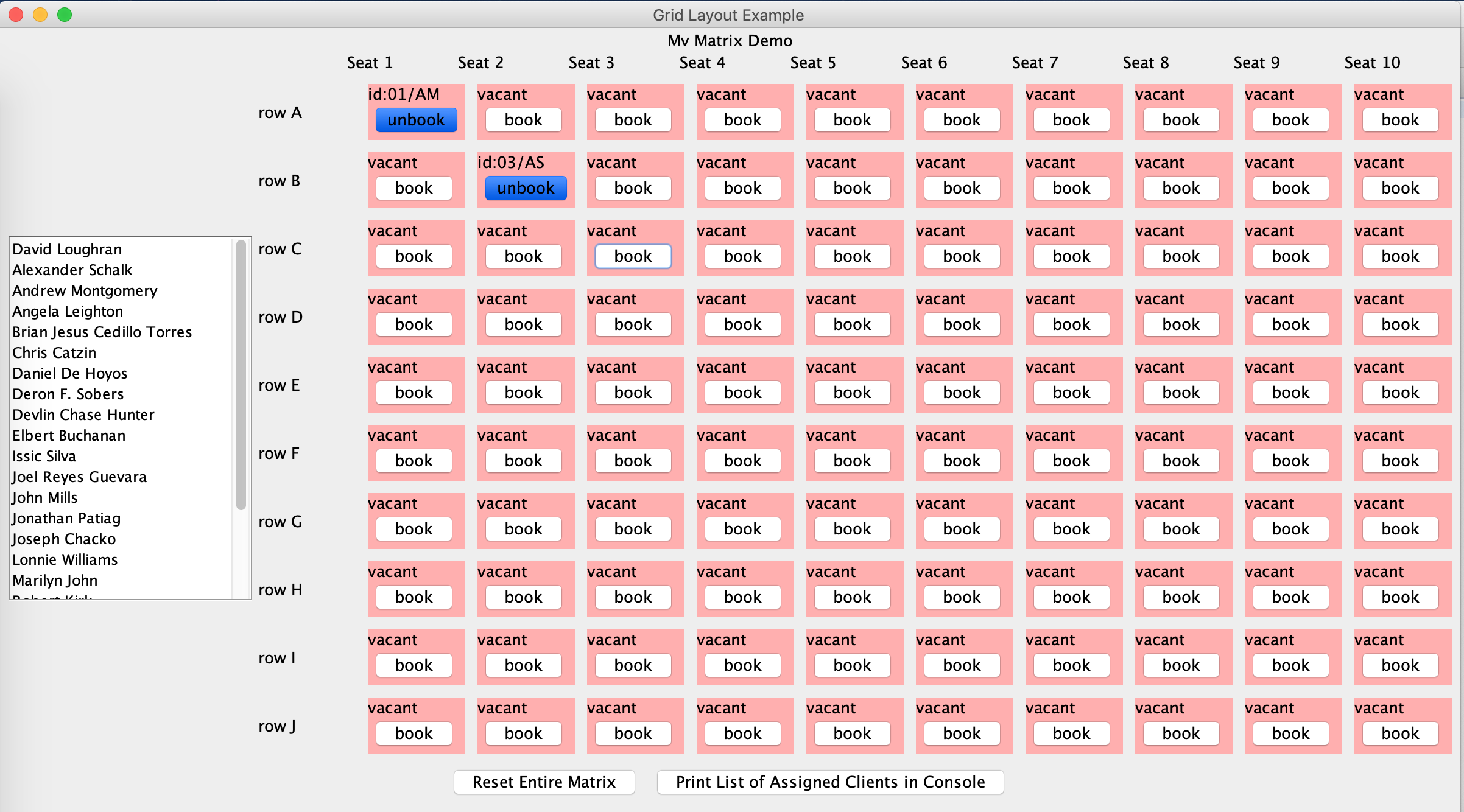
**Figure 1 GUI for Movie Theatre Booking App**

The Clients are initially loaded into a JList inside of a JPanel. For each seat, there’s a JLabel and a JToggleButton. JLabel is initially set as “vacant” when no seat is assigned yet. JToggleButton shows “book” when a seat is unassigned, and switches to “unbook” when a seat is assigned.

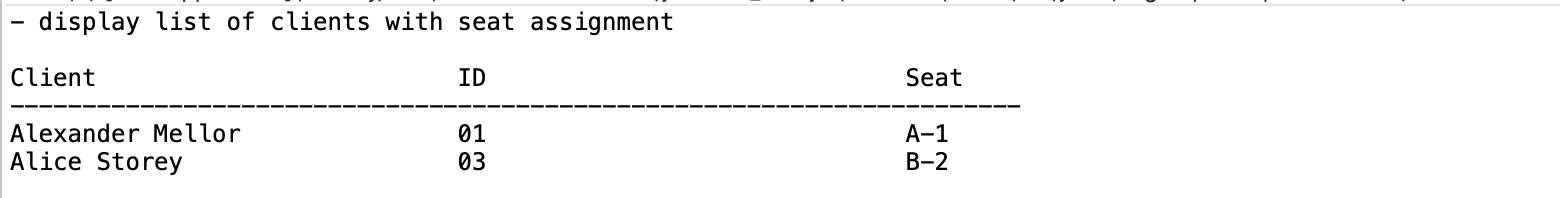


**Figure 2 Booking the Client**

**Figure. 2** demonstrates that three clients are assigned to different seats. In this case, Alexander Mellor, Alice Storey, and David Loughran are assigned to seats in the matrix. Their names are removed from JList when they are assigned. Their id/initial gets displayed with JLabel and toggle button switches from “book” to “unbook”.

** Figure. 3** demonstrates client David Loughran is unbooked from the seating matrix. The seat returns to vacant and “book”, and David Loughran returns to the top of JList.

**Figure 3 Unbooking the Client**

****

**Figure 4 Print Current Seating assignment in Console**

**Figure. 4** shows the result when the button “Print List of Assigned Clients in Console” is clicked. The current seat assignment in the matrix is printed to the console.

****

**Figure 5 Reset Seating Matrix**

**Figure. 5** shows the result when the button “Reset Entire Matrix” is clicked. All assigned clients return back to the top of JList. All seats are emptied.

**Conclusion**

This project further strengthened my ability to program in Object Oriented Programming. In addition, I learned how to use JAVA Swing to develop the graphical user interface. Initially it was a big challenge learning how to use Java Swing, but after reading a ton of online documentations and creating several POCs, I was able to tackle the problem of creating the functional GUI for this app. Finally, through this project, I learned how to dynamically add and remove from a JList by researching online.

The biggest takeaway from this project is learning how to break down the big problem into small sub-problems. Learning how to tackle each small problem at a time is crucial in solving the big problem.

**Potential Improvements**

* Nicer Looking GUI.
* Adding a function to throw an exception or error when another client is trying to book a booked seat.