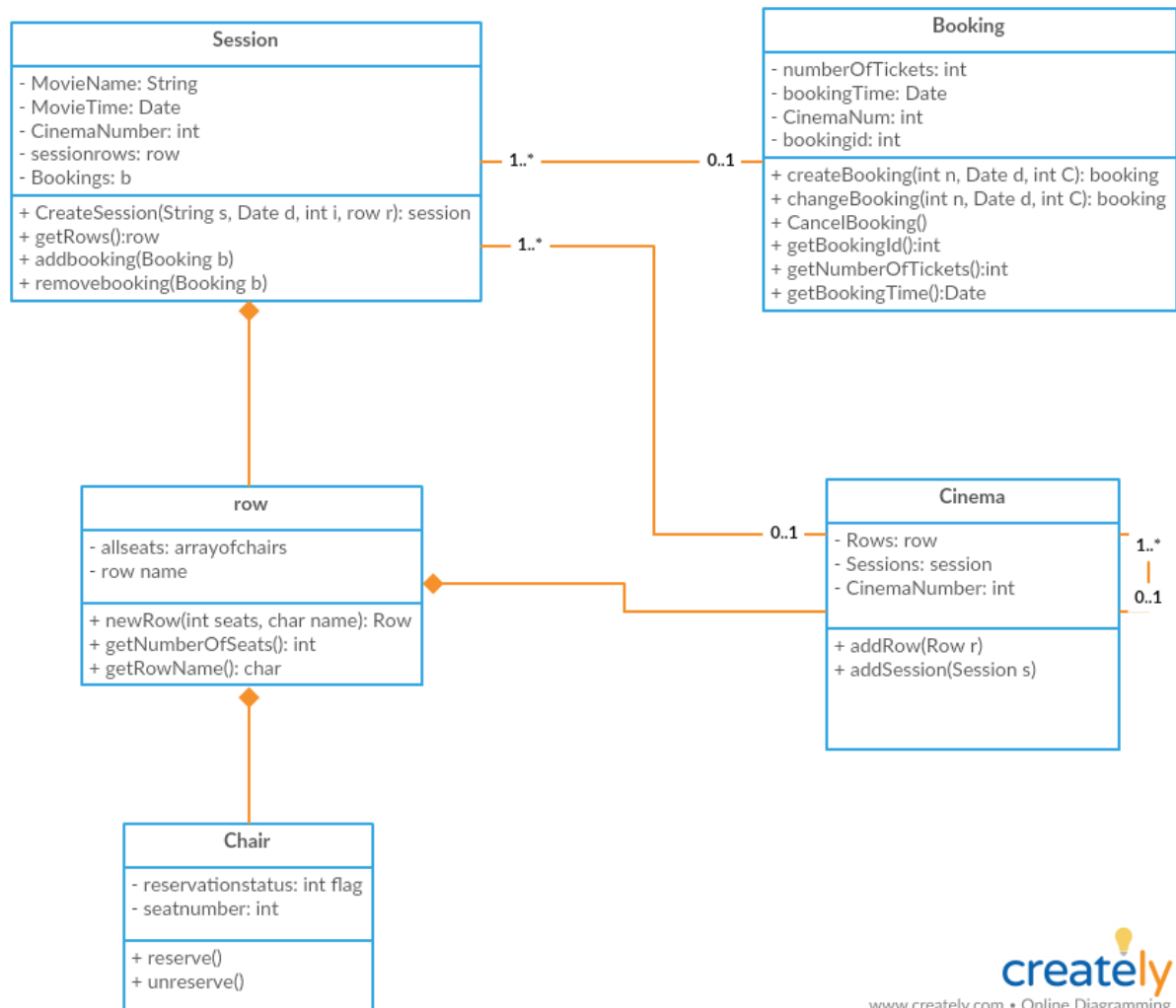# UML Class Diagram

My first UML Class Diagram I made before I started the project which was to try give me a basic idea of how I want to setup my design and relate all my classes, This was made before I started coding and it ended up being a similar design just needed a lot of more functions and also changed some attributes and added some as I went, however this shows how I first thought I wanted to setup the system

Session

- MovieName: String
- MovieTime: Date
- CinemaNumber: int
- sessionrows: row
- Bookings: b

+ CreateSession(String s, Date d, int i, row r): session
+ getRows():row
+ addbooking(Booking b)
+ removebooking(Booking b)

Booking

- numberOfTickets: int
- bookingTime: Date
- CinemaNum: int
- bookingid: int

+ createBooking(int n, Date d, int C): booking
+ changeBooking(int n, Date d, int C): booking
+ CancelBooking()
+ getBookingId():int
+ getNumberOfTickets():int
+ getBookingTime():Date

row

- allseats: arrayofchairs
- row name

+ newRow(int seats, char name): Row
+ getNumberOfSeats(): int
+ getRowName(): char

Cinema

- Rows: row
- Sessions: session
- CinemaNumber: int

+ addRow(Row r)
+ addSession(Session s)

Chair

- reservationstatus: int flag
- seatnumber: int

+ reserve()
+ unreserve()

1..*   0..1
1..*   0..1
0..1   1..*
        0..1

This was edited and fixed up as I made changes to the project, this is the end result after I have finished my entire project, I realised I had to redesign some things and my issue was I visualised a cinema as an object rather than a real thing, when I imagined a cinema as a bunch of theatres with each theatre having a bunch of sessions and people making bookings for a session it clicked with me and I redesigned a lot of things. I also realised I needed a lot more functions than I thought I needed to help navigate

Cinema

- cinemaNumber: int
- rows: ArrayList<Rows>
- session: ArrayList<Session>

+ getSession(): ArrayList<Session>
+ getCinemaNumber(): int
+ addRowToCinema(char c, int n)
+ getSessionIndexBasedOnTime(Date d): int
+ getSessionIndexWithBookingId(int id): int
+ addSession(int n, Date d, String m)
+ addRowToAllSessions(Rows r)

operate

+ runTests(String s)
+ getCinemaNumberIndex(ArrayList<Cinema< cinema, int n)
+ getCinemaWithBookingId(ArrayList<Cinema> c, int n): int
+ printOutcome(String i[ ], ArrayList<Cinema> c)
+ outcomeCancel(String i[ ], ArrayList<Cinema> c)
+ outcomePrint(String i[ ], ArrayList<Cinema> c)
+ outcomeCinema(String i[ ], ArrayList<Cinema> c)
+ outcomeSession(String i[ ], ArrayList<Cinema> c)
+ outcomeRequest(String i[ ], ArrayList<Cinema> c)
+ outcomeChange(String i[ ], ArrayList<Cinema> c)

<<uses>>

CinemaBookingSystem

- main(String args[])

Session

- movie: String
- movieTime: Date
- cinemaNumber: int
- allBookings: ArrayList<bookings>
- sessionRows: ArrayList<Rows>

+ getSessionRows(): ArrayList<Rows>
+ getAllBookings(): ArrayList<bookings>
+ getMovie(): String
+ getMovieTime(): Date
+ addBookingToSession(bookings b)
+ getBookingIdIndex(int i): int
+ printAllBookings()
+ availableRow(int n): int
+ bookSeats(int n, int r)
+ freeSeats(int n, int r, int s)
+ addRowToSession(Rows r)

bookings

- numberOfTickets: int
- bookingTime: date
- cinemaNumber: int
- rowName: String- id: int
- startSeat: int
- rowNumber: int
- startSeat: int

+ getRowName(): String+ getStartSeat(): int
+ getNumberOfTickets(): int
+ getId(): int
+ createBooking(int n, Date d, int i, int cn, ArrayList<Cinema> c, int si, int c): boolean
+ changeBooking(int n, Date d, int i, int cn, ArrayList<Cinema> c, int si, int c): boolean
+ cancelBooking(ArrayList<Cinema>c, int s): boolean
+ checkCinemaAvailable(int c, ArrayList<Cinema> ci): int

<<uses>>

Seats

- seatNumber: int
- reserveStatus: boolean

+ checkReserveStatus():boolean
+ reserveSeat()
+ removeReservedSeat()

Rows

- rowName: String
- chairs: ArrayList<Seats>
- numberOfSeatsInRow: int
- numberOfAvailableSeatsInRow: int

+ getRowName(): String
+ getChairs: ArrayList<Seats>
+ getNumberOfSeatsInRow: int
+ getNumberOfAvailableSeatsInRow(): int
+ addSeatsToRow(int n):int
+ setNumberOfAvailableSeatsInRow(int n)
+ firstFreeSeatInRow():int