

<<control>> CancelTicketController
- view: CancelTicketView - toBeCancelled: Transaction
+ CancelTicketContoller(CancelTicketView, Transaction t) + CancelTicket(Transaction t):void

<<control>> TransactionController
+ view: TransactionView + model: Transaction
+TransactionController(TransactionView theView, ArrayList<Ticket> tickets) + calculatePrice():void + getCardInfo(): void

<<coundary>> MovieView
- view: TerminalView
+ add MovieListener(ActionListener listener): void + MovieView(View theView) + displayMovies(ArrayList<Movie> MovieCatalogue): void + addText(String string): void + clearDisplay(): void + displayErrorMessage(String errorMessage): void

<<control>> LoginController
+ view: LoginView +model: Authenticate
+ LoginController(LoginView theView) + authenticate():void

<<control>> RegisterController
- view: RegisterView - newUser: RegisteredUser
+ RegisterController(RegisterView theView) + createNewAccount():void

<<control>> MovieController
- view: MovieView - availableMovies: ArrayList<Movie> -theatre: Theatre
+ MovieController(MovieView theView, Theatre t) + searchMovie(String title): Movie + getAvailableMovies(): void + selectMovie():void

<<boundary>> RegisterView
+ view: TerminalView
+ add RegisterListener(ActionListener listener): void + RegisterView(View theView) + addText(String string): void + clearDisplay(): void + displayErrorMessage(String errorMessage): void + displayConfirmation(): void

<<control>> ShowingController
- movie: Movie - availableShowings: ArrayList<Showing> - view: AvailableShowingsView
+ ShowingController(AvailableShowingsView theView, Movie theMovie) +selectShowing(): void

<<control>> TheatreController
- view: TheatreView - availableTheatres: ArrayList<Theatre>
+ TheatreController(TheatreView theView) + selectTheatre(): Theatre + getAvailableTheatres(): void + searchTheatre(String name):void

<<boundary>> TheatreView
- view: TerminalView
+ add TheatreListener(ActionListener listener): void + TheatreView(View theView) + displayTheatres(ArrayList<Theatre> availableTheatres): void + addText(String string): void + clearDisplay(): void + displayErrorMessage(String errorMessage): void

<b>&lt;&lt;entity&gt;&gt; User</b>
# email: String # userCredit: ArrayList<MovieCredit> # userID: int # previousPurchases:ArrayList<Transaction>
+ getEmail(): String + setEmail(): void + getUserID():int + setUserID(): void + getUserCredit():void + addCredit(Credit c): void + removeCredit(Credit c): void + getPreviousPurchases(): void + addPurchase(Transaction t): void + removePurchase(Transaction t): void

<b>&lt;&lt;entity&gt;&gt; RegisteredUser</b>
- name: String - address: String - paymentOption: Card - password: String - activeStatus: Boolean - lastPaymentDate: Date
+ getName(): String + setName(String name): void + getAddress(): String + setAddress(String address): void + getPaymentOption(): Card + setPaymentOption(Card card): void + setActiveStatus(): void + getActiveStatus():Boolean +getLastPaymentDate(): Date

<b>&lt;&lt;boundary&gt;&gt; SeatingView</b>
+ view: TerminalView
+ add SeatingListener(ActionListener listener) + SeatingView(View theView) + displaySeating(ArrayList<Seat> seating ): void + addText(String string): void + clearDisplay(): void + displayErrorMessage(String errorMessage)

<b>&lt;&lt;boundary&gt;&gt; LoginView</b>
+ view: TerminalView
+ add LoginListener(ActionListener listener): void + LoginView(View theView) + addText(String string): void + clearDisplay(): void + displayErrorMessage(String errorMessage): void + displayConfirmation(): void

<b>&lt;&lt;boundary&gt;&gt; TerminalView</b>
+ TerminalView() + addButtonListener(ActionListner listener): void + clearDisplay(): void + addText(String string): void + actionPerformed(ActionEvent e): void + getStringInput():String + getIntegerInput(): int + displayErrorMessage(String errorMessage)

<b>&lt;&lt;boundary&gt;&gt; AvailableShowingsView</b>
+ view: TerminalView
+ add ShowingListener(ActionListener listener): void + ShowingView(View theView) + displayShowings(ArrayList<Showing> showingCatalogue): void + addText(String string): void + clearDisplay(): void + displayErrorMessage(String errorMessage): void

<b>&lt;&lt;boundary&gt;&gt; TransactionView</b>
+ view: TerminalView
+ add TransactionListener(ActionListener listener): void + TransactionView(View theView) + displayTransaction(Transaction t): void + addText(String string): void + clearDisplay(): void + displayErrorMessage(String errorMessage): void

<b>&lt;&lt;boundary&gt;&gt; CancelTicketView</b>
+ view: TerminalView
+ add CancelTicketListener(ActionListener listener): void + CancelTicketView(View theView): void + displayTransactionHistory(ArrayList<Transaction> transactionHistory): void + addText(String string): void + clearDisplay(): void + displayErrorMessage(String errorMessage): void

<b>&lt;&lt;control&gt;&gt; SeatingController</b>
- showing: Showing - view: SeatingView - numberOfTickets: int
+ SeatingController(SeatingView theView, Showing theShowing) + selectSeats():void

<b>&lt;&lt;entity&gt;&gt; Seat</b>
- seatId: int - reserved: boolean - seatNumber: int
+ isSeatAvailable(): boolean + reserveSeat(): void

<b>&lt;&lt;entity&gt;&gt; Showing</b>
- showingId: int - movie: Movie - theatre: Theatre - showTime: Date - ticketPrice: double
+ getMovie(): Movie + setMovie(Movie movie): void + getTheatre(): Theatre + setTheatre(Theatre theatre): void + getShowTime(): Date + setShowTime(Date showTime): void

<b>&lt;&lt;entity&gt;&gt; Movie</b>
- movieId: int - title: String - description: String - showings: ArrayList<Showing>
+ getTitle(): String + setTitle(String title): void + getDescription(): String + setDescription(String description): void + getShowings(): ArrayList<Showing> + addShowing(Showing showing): void + removeShowing(Showing showing): void

<b>&lt;&lt;entity&gt;&gt; Theatre</b>
- theatreId: int - showings: ArrayList<Showing> - seats: ArrayList<Seat>
+ getShowings(): ArrayList<Showing> + addShowing(Showing showing): void + removeShowing(Showing showing): voi + getSeats(): ArrayList<Seat>

<b>&lt;&lt;entity&gt;&gt; Transaction</b>
- transactionId: int - user: User - tickets: ArrayList<Ticket> - totalCost: double - purchaseDate: Date - paymentCard: Card
+ getUser(): User + setUser(User user): void + getTickets(): ArrayList<Ticket> + addTicket(Ticket ticket): void + removeTicket(Ticket ticket): void + getTotalCost(): double + setTotalCost(double cost): void + getPurchaseDate(Date purchaseDate): void + getCard(): Card + setCard(Card card): void

<b>&lt;&lt;entity&gt;&gt; Ticket</b>
- ticketId: int - showing: Showing - seat: Seat
+ getShowing(): Showing + setShowing(Showing showing): void + getSeat(): Seat + setSeat(Seat seat): void

<b>&lt;&lt;entity&gt;&gt; Card</b>
- cardId: int - cardHolderName: String - cardNumber: int - expiryMonth: int - expiryYear: int - cvv: int
+ getCardHolderName(): String + setCardHolderName(String name): void + getCardNumber(): int + setCardNumber(int cardNumber): void + getExpiryMonth(): int + setExpiryMonth(int expiryMonth): void + getExpiryYear(): int + setExpiryYear(int expiryYear): void + getCVV(): int + setCVV(int cvv): void

<b>&lt;&lt;entity&gt;&gt; MovieCredit</b>
- movieCreditId: int - creditCode: String - expiryDate: Date - amount: double
+ getCreditCode(): String + setCreditCode(String creditCode): void + getExpiryDate(): Date + setExpiryDate(Date expiryDate): void + getAmount(): double + setAmount(double amount): void

<b>&lt;&lt;entity&gt;&gt; Authenticate</b>
- registeredUser: RegisteredUser
+ checkName(): Boolean + checkPassword(): Boolean

<b>&lt;&lt;entity&gt;&gt; EmailProvider</b>
- ticket: Transaction - seat: Seat
+ getTicketInfo(): Transaction + sendUserEmail(): Transaction + sendTicketAndReceipt(): void +getMovieAnnounment(): void + sendMovieAnnounment(): void