UML Class Diagram Description- BEM Theater System

The UML Class Diagram for the BEM Theater System models the structure of a movie ticketing platform. It defines the main classes, their attributes and methods, and how they relate to each other. This diagram serves as a blueprint for implementing features like ticketing purchasing, showtime scheduling, user management, and administrative control.

UserAccount

Represents a costumer who uses the system

Attributes:

- id: Unique user identifier of the customer (private)
- name: Full name of customer (private)
- email: Contact email of customer (private)
- password: Encrypted login password (private)
- paymentMethods[]: List of saved payment options (private)
- loyaltyPoints: Points earned through purchases (private)

Methods:

- validate(): Checks login credentials (private)
- refund(): Requests a refund (private)
- sendByEmail(): Sends confirmation or ticket via email (private)

Admin (Inherits from UserAccount)

Represents a staff member with special privileges

Additional Methods:

- setShowtimes(): Staff member will be able to set showtimes for movies (private)
- editTheatreAssignment(): (private)
- viewLogs(): (private)
- overrideTransaction(): Rejects transaction (private)
- initiateRefund(): Staff member will be able to give customer a refund (private)

Movie

Represents a film available in the system

Attributes:

- title: Name of the movie (private)
- duration: Length of movie in minutes (private)
- rating: Content rating (private)
- reviews[]: List of critic reviews (private)

Methods:

- getShotimes(): Returns scheduled showings (public)
- fetchReviews(): Gets review data (public)

Showtime

Represents a scheduled screening of a movie

Attributes:

- id: Unique identifier (private)
- startTime: Date and time of the showing (private)
- movie: Linked movie (private)
- theatre: Theatre where its shown (private)
- availableSeats[]: List of seats not yet reserved (private)

Methods:

- reserveSeat(seatId): Reserves a specific seat (private)

Theatre

Represents a physical cinema location

Attributes:

- id: Unique theatre ID (private)
- location: Address or city of a specific theater (private)
- seats[]: All seats in the theatre (private)

Methods

- getAvailableSeats(): Returns unreserved seats (public)

- getDeluxeSeats(): Returns seats of type ‘Deluxe’

Seat

Represents an individual seat in a theatre

Attributes:

- id: Seat identifier (private)
- type: Regular or Deluxe (private)
- isAvailable: Whether the seat is free or not (private)

Methods:

- reserve(): Marks seat as taken (public)
- release(): Frees the seat (public)

Ticket

Represents a purchased ticket linked to a user and a showtime

Attributes:

- ticketId: Unique identifier (private)
- owner: Associated UserAccount (private)
- showtime: Linked Showtime (private)
- seat: Linked Seat (private)
- isNFT: Whether it’s a digital token (private)

Methods:

- generateQR(): Creates QR code for ticket (private)
- sendToUser() Sends ticket to the user (private)

PaymentMethod

Represents a stored payment option

Attributes:

- type: Payment method (e.g. Paypal, Credit Card) (private)
- details: Encrypted or tokenized info. (private)

Review

Represents a review from an external source

Attributes:

- source: e.g. Rotten Tomatoes or IMDB (private)
- score: Numerical rating (private)
- quote: Critic comment (private)

Relationships

- UserAccount has max amount of 20 Tickets
- Ticket links to one Showtime and one Seat
- Showtime links to one Movie and one Theatre, and to many Seats.
- Theatre contains many Seats (composition), either regular or deluxe depending on the theatre
- Movie aggregates Reviews
- UserAccount aggregates PaymentMethods
- Admin is a subclass of UserAccount with additional methods

Updates

From July 22, 2025, to July 29, 2025, what has been updated to both UML diagram and description was:

- Description of classes and methods types (private or public)
- More visual links on diagram
- Organization of both diagram and description
- Additional information on relationships

This description covers the complete structure and purpose of the system's classes and their interactions, fulfilling the requirements for a full UML Class Diagram