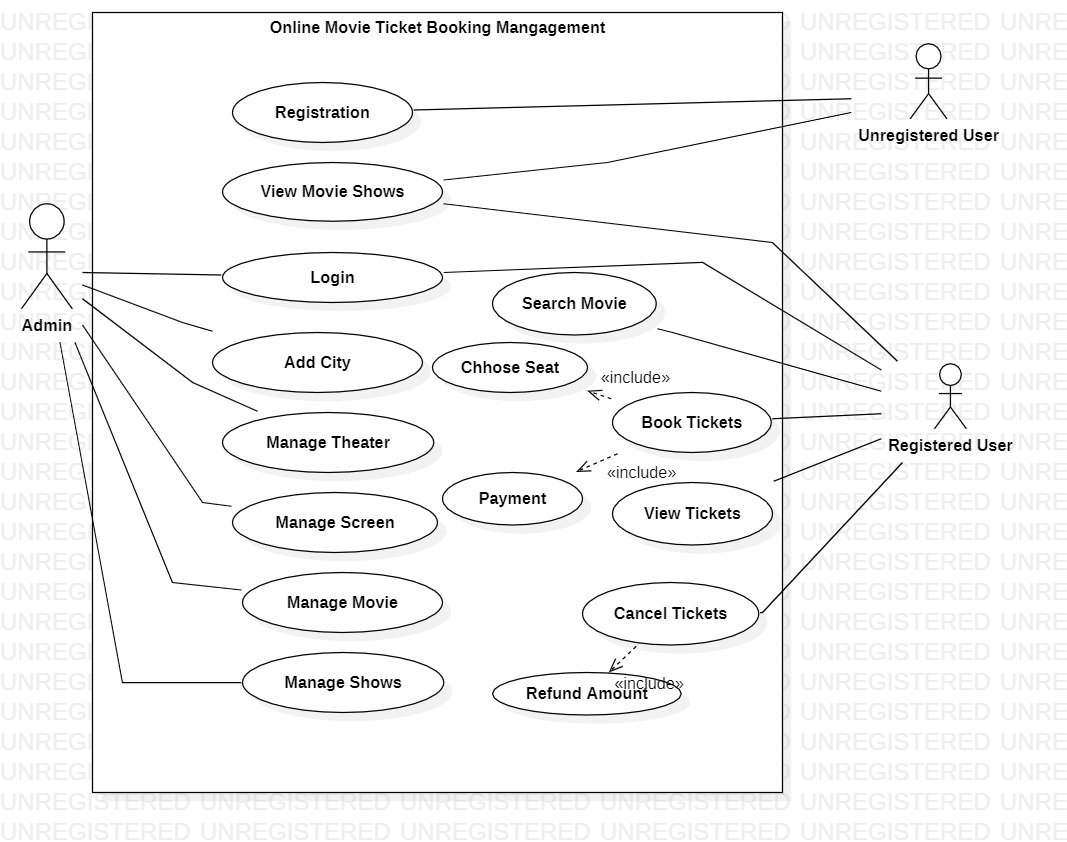
**ONLINE MOVIE TICKET BOOKING**

**INTRODUCTION**

This is an online movie ticket system having an admin role and a customer role where admin has access to add city, the theatres in the city, the movies each theatre will showcase and the show timings for each movie. The customer can be an unregistered user or a registered user. Unregistered user can register itself or can view movie shows without registering while the registered user can view movie shows and book tickets for the particular show by paying online. Registered users will have the option to cancel its booking in which case he will get a refund on some specific terms. Users can select the language of movie and view the seat map and choose seats for themselves.

**Use Case**

****

**Modules**

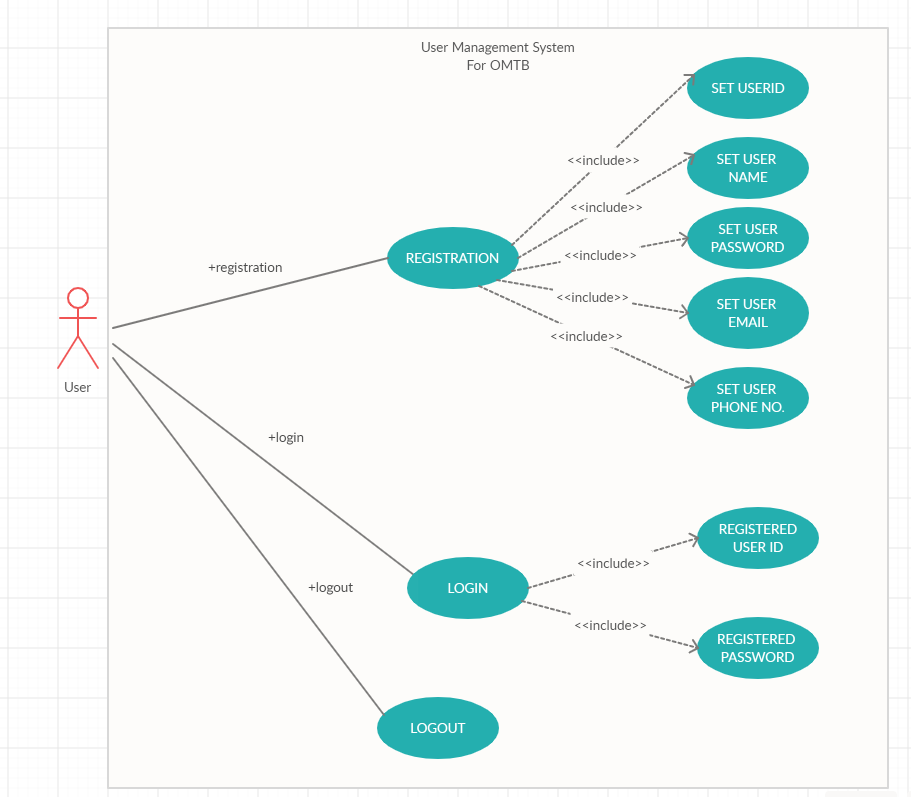
**USER/ADMIN MANAGEMENT SYSTEM**

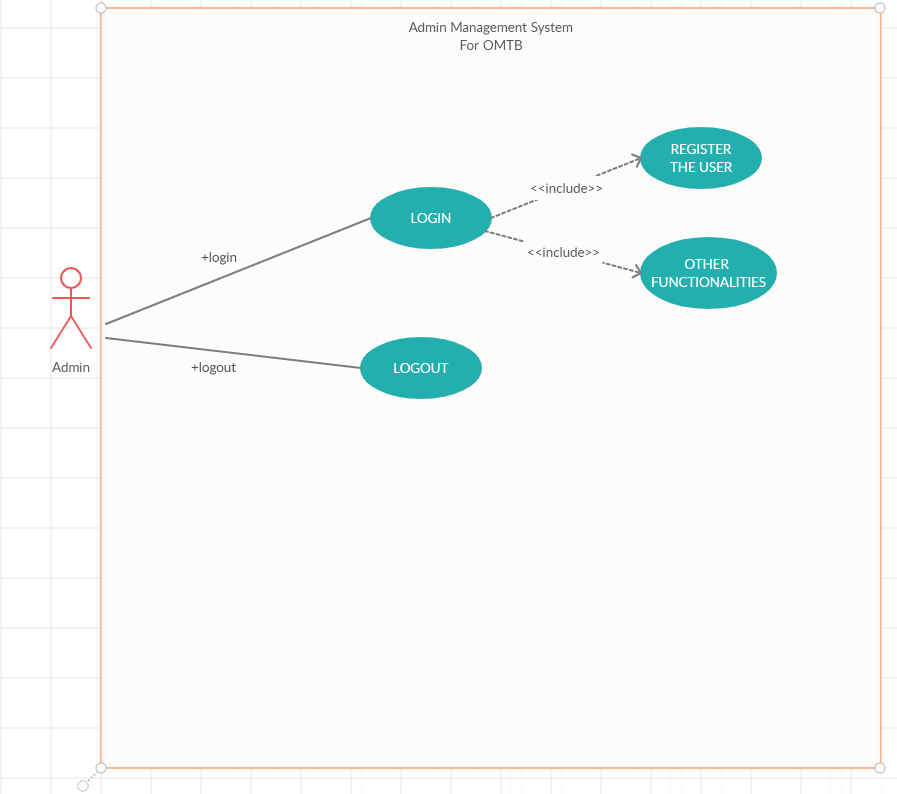
**FUNCTIONALITY:-**

* Register User
* Sign in User/Admin
* Sign out User/Admin

**CLASS ATTRIBUTES:-**

* userId : String
* userName : String
* userPassword: String
* userEmail : String
* userPhone : Integer
* 4) adminId : String
* adminPassword : String





**Theater Management**

**Functionalities:-**

* Add Theater
* Delete Theater
* Search Movie

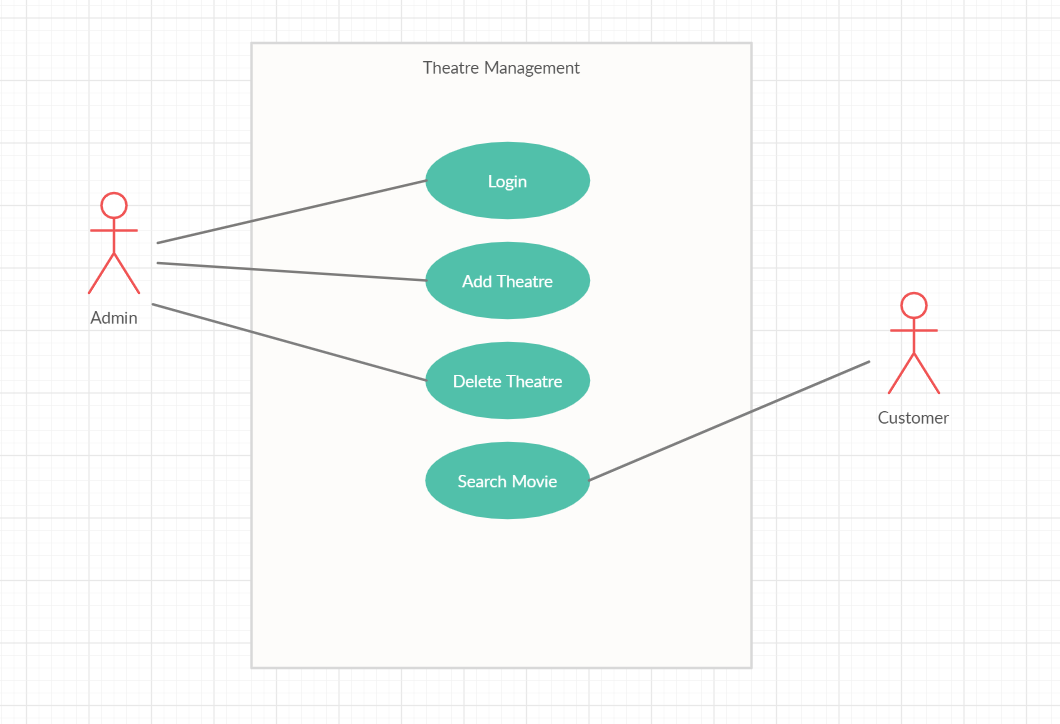
**Class: Theater**

**Attributes:**

* theaterId: Integer
* theaterName: String
* theaterCity: String
* movies: Movie[\*]
* listOfScreens: Screen[\*]
* managerName: String
* managerContact: String

**Validations:-**

* Theater ID Validation
* Theater Name Validation
* City Validation

**Use Case Diagram:-** 

**Movie Management System**

**Functionalities:-**

* Add Movie
* Delete Movie
* Show Movie

**Class: Movie**

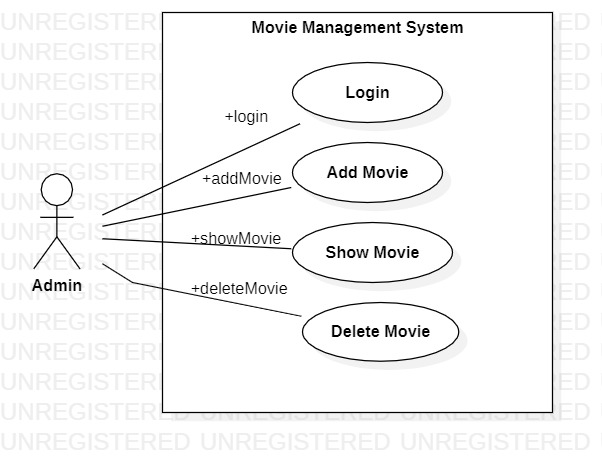
**Attributes**

* movieId: Integer
* movieName: String
* movieGenre: Show[\*]
* movieDirector: String
* movieLength: Integer
* languages: String[\*]
* movieReleaseDate: LocalDate

**Validations:-**

* Name Validation
* Date Validation
* Language validation

**Use Case Diagram:-**



**Screen Management System**

**Functionalities:-**

* Add Screen
* Delete Screen
* Show Screen

**Class: Movie**

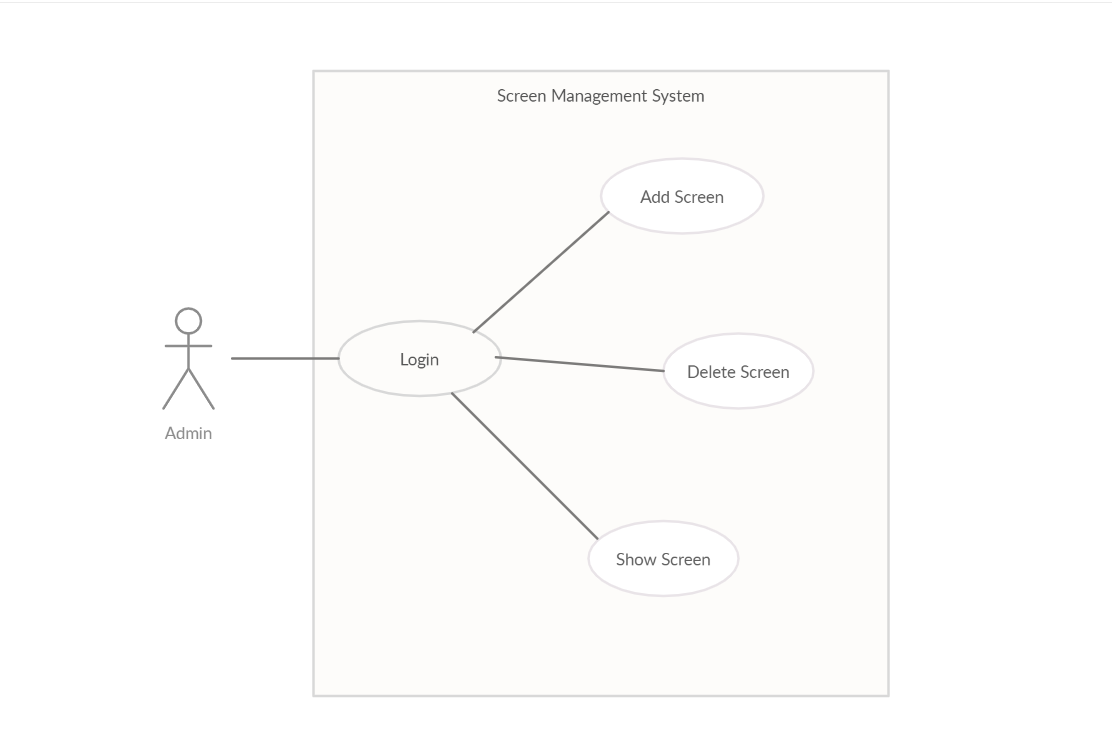
**Attributes**

* screenId: Integer
* screenName: String
* movieEndDate: Date
* rows: Integer
* column: Integer
* threaterId: Integer

**Validations:-**

* Name Validation
* Existence Validation

**Use Case Diagram:-**



**Show management:**

**Class show:**

* showId: Integer, 5 digit
* showStartTime: Time, cannot be null
* showEndTime: Time, cannot be null
* movieName: String, cannot be null
* screenId: Integer, cannot be null
* theaterId: Integer, cannot be null

**validation:**

* Showed validation
* Name validation

**Functionality:**

* Add show
* Delete show

**Use Case diagram:**

admin

**Booking Management System**

**Functionalities:-**

* View Movie Shows
* Choose Seat
* Calculate Total Cost
* Make Payment
* Book Tickets
* View Tickets
* Cancel Tickets
* Refund Amount

**Class: Booking**

**Attributes**

* bookingId: Integer
* movieId: Integer
* showId: Integer
* showRef: Show
* bookingDate: LocalDate
* transactionId: Integer
* totalCost: Double
* seatList: Seat[\*]
* ticket: Ticket

**Class: Ticket**

**Attributes**

* ticketId: Integer
* noOfSeats: Integer
* seatName: String[\*]
* bookingRef: Booking
* ticketStatus: Boolean
* screenName: String

**Validations:-**

* Booking Date should be either present day or previous not the future.
* BookingId will be 10 digit with customerId, theaterId , movieId , showId concatenated.
* Ticket Id will be min 8 digit and will have booking and transactionId combination.
* If seat is not available then ticket will not booked
* Movie show booked Date and time should be either present Date or future.

**Use Case Diagram:-**

