

UML Class Description

User:

Functions:

When a user is logged into the account they will have options like `closestCinema()`, `searchMovies()`, `checkShowtime()`, and `bookTicket()`. If they do not want to visit the main site in Coronado, they will have the option to click the `closestCinema()` function and be navigated to the other cinemas that are showing the desired movie. If the user is trying to search a specific movie they can click the `searchMovies()` function and type in the movie they are trying to watch and it will direct them to a different page with the search results of what they typed in. Once the user has found a movie they would like to watch they can navigate to the `checkShowtime()` function where they will be able to see times when the movie will be played to see which time meets their availability. Once they have found a movie and picked a showtime, they can finalize everything by clicking the `bookTicket()` function.

accountInfo:

Attributes:

In the user's `accountInfo` they will need to provide their `name(String)`, `email(String)`, `password(String)`, and `CCinfo(Int)` when they are making their accounts. The user will need to provide their first and last name. Then provide their personal emails as well as a password that meets the requirements in order to protect the users account and make sure they create a strong enough password so people cant access their accounts easily. When the account is made and the user finds a movie they want to watch at the cinema the user will have to provide their CC info in order to purchase the tickets.

Functions:

The functions for the `accountInfo` class are `login()`, `register()`, `editProfile()`, and `resetPassword()`. When the user is not logged into their account or they log onto their account on another device they will have to use the `login()` function to do so. If the user is a first time user they will be prompted to `register()` in order to purchase a ticket. When a user wants to change something on their account they will have to use the `editProfile()` function. When a user forgets their login password they can use the `resetPassword()` function to reset their password so they can login.

Ticket:

Attributes:

For the `Ticket` attributes they are `ticketId(int)`, `seatNumber(String)`, and `price(float)`. The tickets will have a `TicketId` on them to identify what movie it is for. The ticket will include the user's `seatNumber` so the user will know where to sit. To watch the movie the user will have to purchase the ticket and it will display the ticket price.

Functions:

The functions for the `Ticket` class is `bookTicket()` and `cancelTicket()`. When the user finds a movie that they want to book ticket(s) for they can use the `bookTicket()` function for that, and if

they book it and later want to cancel the ticket within an appropriate time frame then they can use the `cancelTicket()` function.

Movie:

Attributes:

For the Movie the attributes are `movieId(int)`, `title(String)`, `genre(String)`, `cast(String)`, and `director(String)`. Each movie will have the `movieId` which is a list of numbers special to that movie to help users navigate the movie better. The title is the name of the movie. The genre is which genre the movie is (ex: Comedy, Horror, Drama, etc.). The cast will be shown so the customers can see who is in the movie. The credits for the director will be shown on the movie credits as well. Lastly, the `movieRating` will be shown to notify the future customer on the age limit for the movie.

Functions:

The functions for the Movie class are `getMovieDetails()`, `addShowtime()`, and `removeShowtime()`. The `getMovieDetails()` function will give a brief summary of the movie for the user and will also give them information on the movie for the user (ex: cast, genre, director, etc.). The `addShowtime()` function will be for the people who run the website so they can add future showtimes that will broadcast the movie. Same thing with the `removeShowtime()` function but instead of adding a showtime it will be removing an existing one.

Showtime:

Attributes:

The attributes for the showtime class are `showtimeId(int)`, `time(Date Time)`, `availableSeats(map<int, bool>)`. The `showtimeId` is also a list of numbers special to the specific movie showtime. The time will be running off of real world time and will manage when movies start and what times are available/not available anymore. The `availableSeats` attribute will show how many seats are available for that specific movie and showtime and it will make sure that if there are none available anymore that the website doesn't sell more seats than available.

Functions:

The functions for the Showtime class are `getAvailableSeats()`, `reserveSeats()`. The `getAvailableSeats()` function will pull up a window where it will show all the seat(s) in the theater room and it will show which ones are/aren't available. Once the user has found which seat(s) they want to reserve they can select them then click the `reserveSeats()` function to finalize them and make them unavailable for everyone else.

makePayment:

Attributes:

The attributes for the makePayment class are `transactionId(int)`, `amount(float)`, and `paymentMethod(String)`. The `transactionId` will be a ID special to that transaction that will be in the system just in case if the customer wants a refund or something goes wrong on our end. The amount attribute will display how much the user owes at the end of their transaction. Lastly,

the paymentMethod attribute will display different ways the user can pay for their balance at the end.

Functions:

The functions for the makePayment class are processPayment() and refundPayment(). The processPayment() function will prompt the user that their payment is being processed and after a very short time the payment will go through and will redirect them to another page with their payment info and a transactionId to be used like a receipt just in case anything goes wrong. The refundPayment() function will be used if the user for any reason would want a refund before the set deadline period and they can use that option to do so, if applicable.