1. User selects usertype / User Login

When user opens the application, the application prompts the user to select the user type. If a user registered before, the application has his information recorded, and he will be able to proceed with usecase 2 or usecase 3. If the user is not registered before, he is an ordinary user and will proceed with usecase 3.

 Objects:

-RegisteredUser

-OrdinaryUser

-Application

Potential Operations:

* selectUsertype()

 2. Registered User browse upcoming movies

The user selects “browse movie” drop-down to browse for upcoming movies and selects a movie.

Objects:

-User

-MovieList

-Movie

Potential Operations:

* displayReservationOnlyMovie()
* selectMovie()

3. Ordinary user searches and selects a movie

The user enters the app and searches/browse for movies available, then select a movie that they wants to watch.

Objects:

-User

-MovieList

-Movie

Potential Operations:

* displayRegularMovie()
* selectMovie()

4. User selects for a theatre

After selecting a movie, the user selects the theater that is available around his/her location.

Objects:

-User

-movie

-theater

Potential Operations:

* theater.toString()
* selectTheater()

5. User selects movie show-time

After selecting the theater the user browse a list of showtime and corresponding rooms for the movie that they selected. The user then selects a showtime for a movie that he wants to watch at.

Objects:

-User

-Movie

-Theater

-Showtime

Potential Operations:

* selectShowtime()
* displayShowtimes()

6. View and select the seat

The user selects the seat that he/she wants to sit in for the movie. The application displays the GUI for all the seats in the theatre, with available and unavailable seats differentiated visually.

Objects:

-User

-movie

-theatre

-application

-seatingGUI

-seat

Potential Operations:

* selectSeat
* seatingGUI.display()

7. Purchase the ticket and check out

After the user selects the seat, the application asks the ordinary user to enter user-information, enter card-information. If the user is a registered user the previous information requested is pre-filled. When he submits the payment, the system confirms the completion of the process. The system sends the ticket and a receipt to his/her email address.

Objects:

-Ticket

-application

-User

-RegisteredUser

-OrdinaryUser

-payment

-userInformation

-cardInformation

-financialInstitute

-receipt

-userType

Potential Operations:

* authenticatePayment()
* processPayment()
* emailConfimation()
* selectUserType()
* getUser()
* collectUserInformation()

8. User cancels the ticket

1)The user successfully refunded the ticket if he refunds up to 72 hours prior to the showtime. If the user is the ordinary user, he receives a coupon with 15% administration fee for future ticket purchase. If the user is a registered user, he will not be charged for administration fee and will get the full credit. However, the credit has a maximum of 1-year for expiration.

2)The user wanted to refund the ticket, but he was not able to successfully refund it because he is an ordinary user it is within 72 hours of the show.

Objects:

-Ticket

-Showtime

-OrdinaryUser

-RegisteredUser

-Coupon

-Cancellation

-Receipt

Potential Operations:

* refund()
* requestRefund()

9. Registered user gets charged for $20 annual fee

The registered user is requested to pay annual account fee ($20) every year.

Objects:

RegisteredUser

Potential Operations:

-payAnnualFee()