MTS (Movie Ticketing System)- MPP Project

■ Background

Movie theatre is a place where movies are screened. Different movie can be screened in same theatre or in different theatre in different timeline. Customer (i.e. movie viewer) may come to theatre and watch movie. For watching movie, they have to buy tickets. Ticket includes seat number and have price which they have to pay. After getting ticket they go to relative theatre to watch movie. If we can design the digital ticketing system, it will not only be beneficial to theatre staffs and owner to track their system but can facilitates customer to check status and get tickets easily.

■ Problem Statement:

There are three major actors in this system: administrator, normal user and customer. Administer is responsible for adding normal users, adding movies, allocating movie to the theatre, fixing ticket rates and editing for all the operations. Normal users are responsible for taking customer order, checking seat availability and issuing movie tickets to customer.   Customer ask to normal user for the movie ticket and get the ticket if available.

All users have their unique user id, first name, last name, address and user role. vUser role tracks the user for their operations.

Theatre has theater name and theater location and list the information of movies which are screening.

Movie has name, genre and length and information about theatre allocated. Movie should have information about customer feedback.

Show time has show type and show time. It also contains information of theatre in which the show time is scheduled.

Ticket has information about ticket class, ticket price, theatre info and seat numbers, price and customer info.

One theatre may have different shows which can be assigned to different movies. Theatre also have information about seats which have its status.
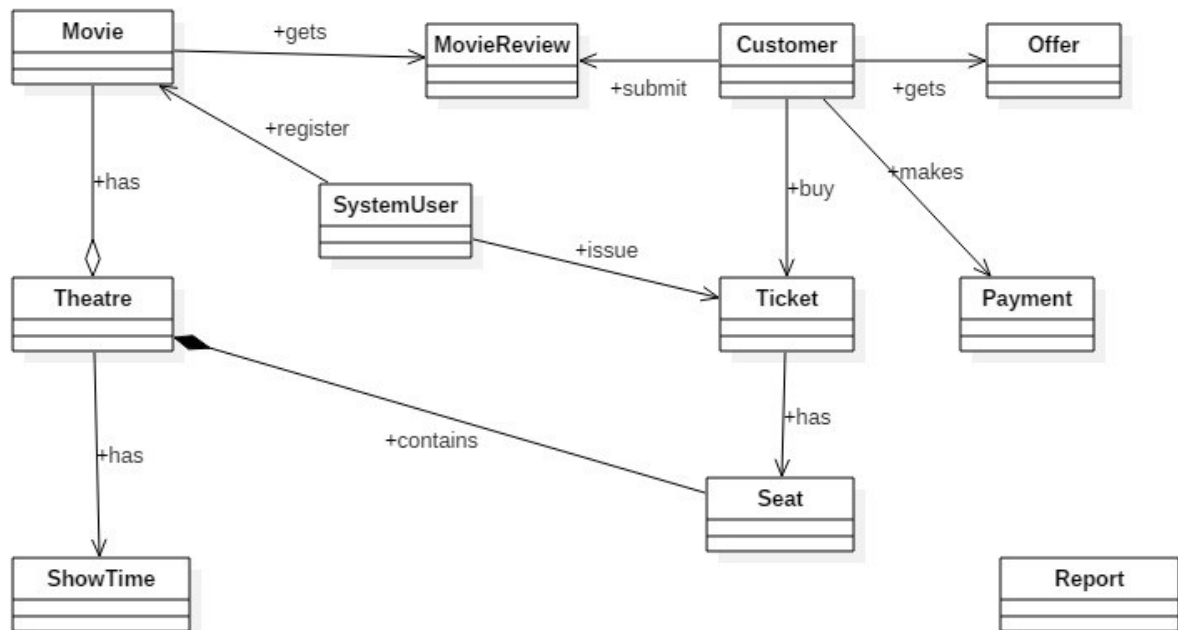
Seat status of theatre of assigned movie is updated during the issue of tickets.
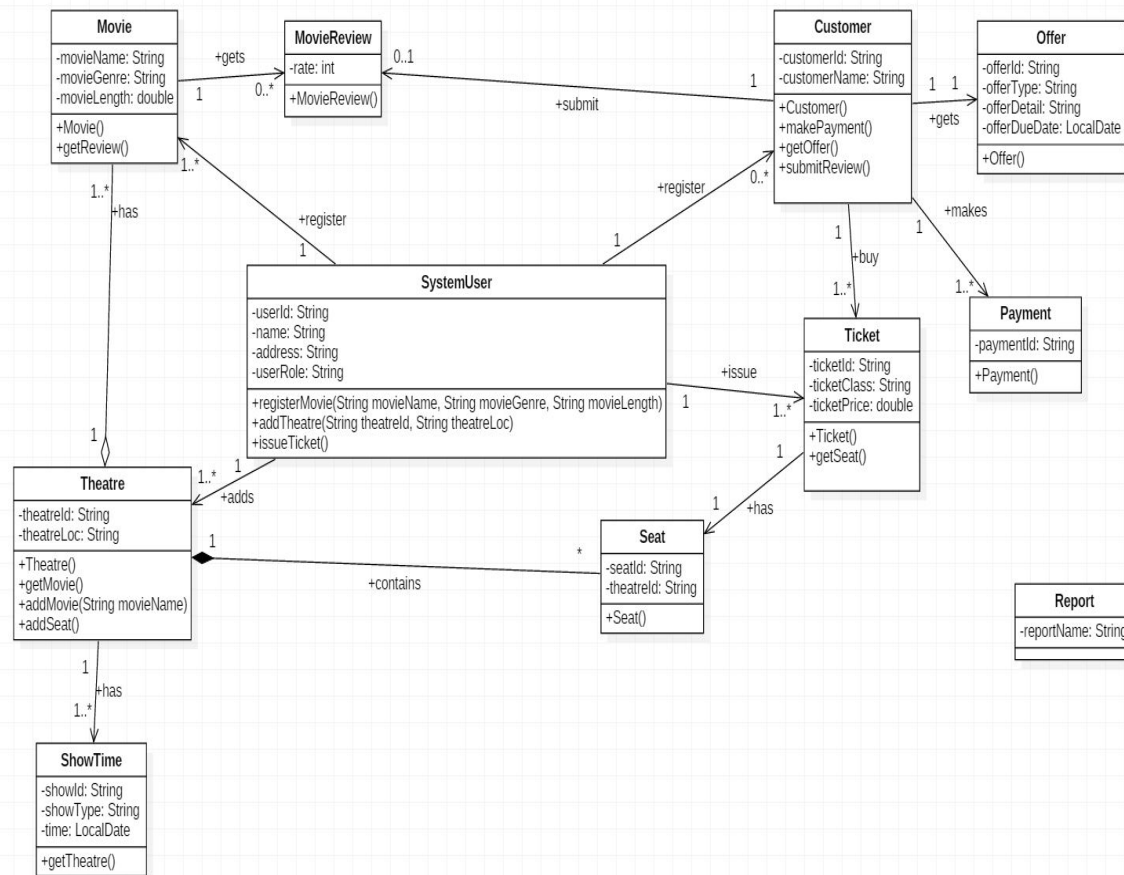
◼Identified Classes

- SystemUser

- Theatre

- Movie

- ShowTime

- Ticket

- Seat

- Customer

- MovieReview

- Payment

- Offer

- Report

◼Initial Class Diagram

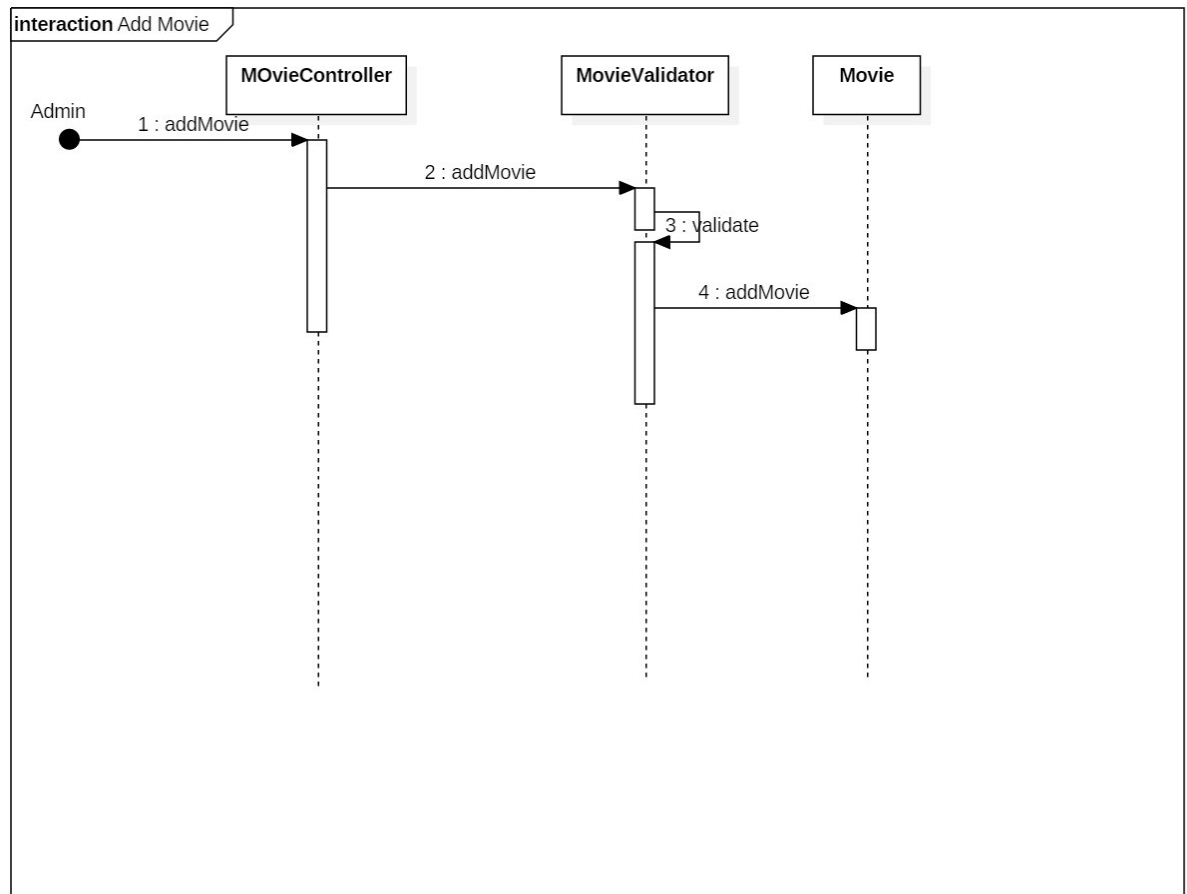Initial class diagram is shown below:

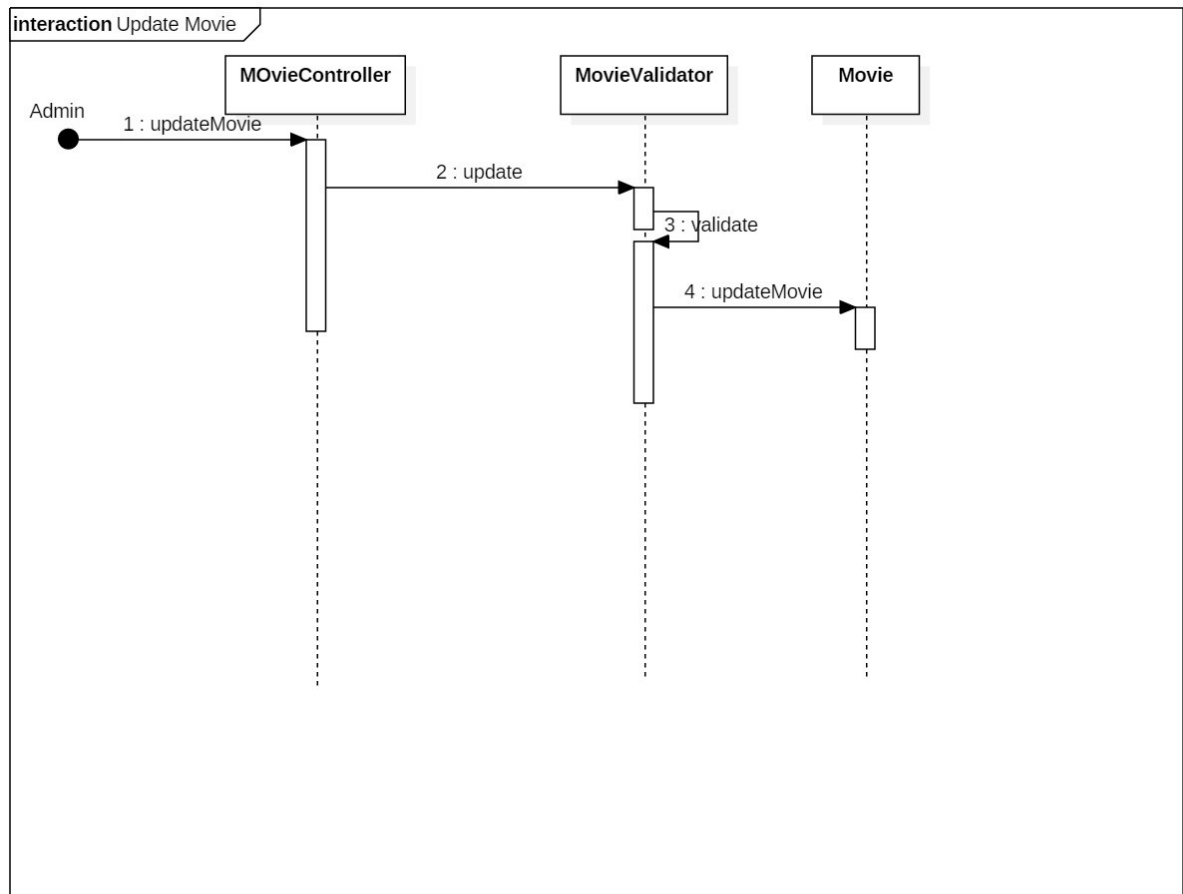Class diagram with attributes, methods and association:

Movie
-movieName: String
-movieGenre: String
-movieLength: double
+Movie()
+getReview()

MovieReview
-rate: int
+MovieReview()

Customer
-customerId: String
-customerName: String
+Customer()
+makePayment()
+getOffer()
+submitReview()

Offer
-offerId: String
-offerType: String
-offerDetail: String
-offerDueDate: LocalDate
+Offer()

+gets
1
0..*
0..1
+submit
1
+register
1   1
+gets
+makes

SystemUser
-userId: String
-name: String
-address: String
-userRole: String
+registerMovie(String movieName, String movieGenre, String movieLength)
+addTheatre(String theatreId, String theatreLoc)
+issueTicket()

+register
1
1..*
+has
1..*
1

Payment
-paymentId: String
+Payment()

+buy
1
1..*
1..*

Ticket
-ticketId: String
-ticketClass: String
-ticketPrice: double
+Ticket()
+getSeat()

+issue
1
1..*
1

Theatre
-theatreId: String
-theatreLoc: String
+Theatre()
+getMovie()
+addMovie(String movieName)
+addSeat()

1..*
+adds
1

Seat
-seatId: String
-theatreId: String
+Seat()

+has
1
*
+contains
1

Report
-reportName: String

1
+has
1..*

ShowTime
-showId: String
-showType: String
-time: LocalDate
+getTheatre()

■Sequence Diagram (major Functions only)

5.1 Add Movie

interaction Add Movie

| | MOvieController | MovieValidator | Movie |

Admin

1 : addMovie

2 : addMovie

3 : validate

4 : addMovie

5.2 Update Movie

interaction Update Movie

Admin

MOvieController   MovieValidator   Movie

1 : updateMovie

2 : update

3 : validate

4 : updateMovie

5.3 Allocate Theatre

interaction Allocate Theatre

Admin — 1 : addMovie → MOvieController
MOvieController — 2 : addMovie → MovieValidator
MovieValidator — 3 : validate → MovieValidator
MovieValidator — 4 : addMovie → Movie
Movie — 5 : allocateTheatre → Theatre
Theatre — 6 : allocateShowTime → ShowTime

5.4 Check Seat Status

interaction Check Seat Status

| Seat Controller | Seat | Theatre | ShowTime | Movie |

Normal User

1 : CheckSeatStatus

2 : getTheatre

3 : getMovie

4 : getShowTime

5 : checkSeatStatus

5.5 Get Show Time

5.6 Book Ticket

interaction Book Ticket

Seat Controller | Seat | Theatre | ShowTime | Movie | Ticket

Normal User
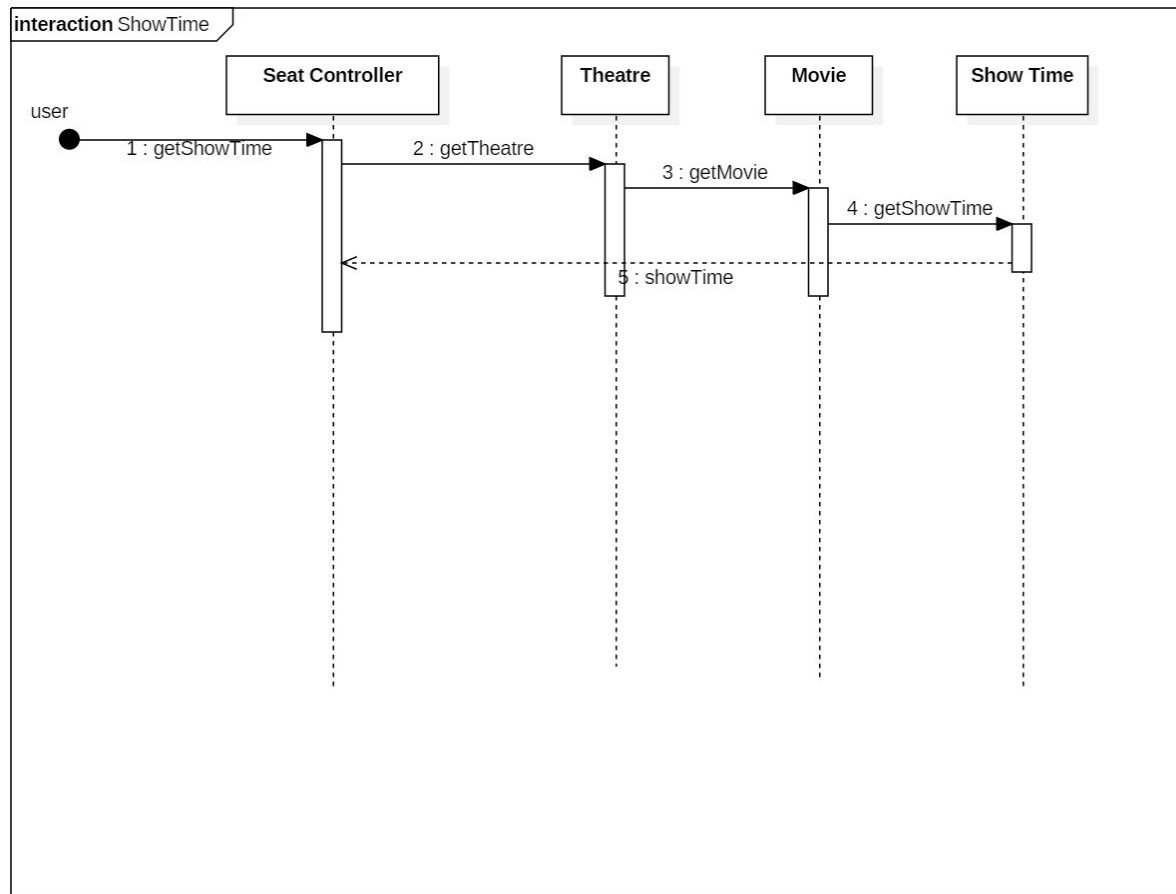
1 : CheckSeatStatus
2 : getTheatre
3 : getMovie
4 : getShowTime
5 : checkSeatStatus
6 : bookTicket

5.7 Submit Review

interaction Movie Review

| CustomerController | Customer | Movie | MovieReview |

Customer

1 : movieReview

2 : getCustomer

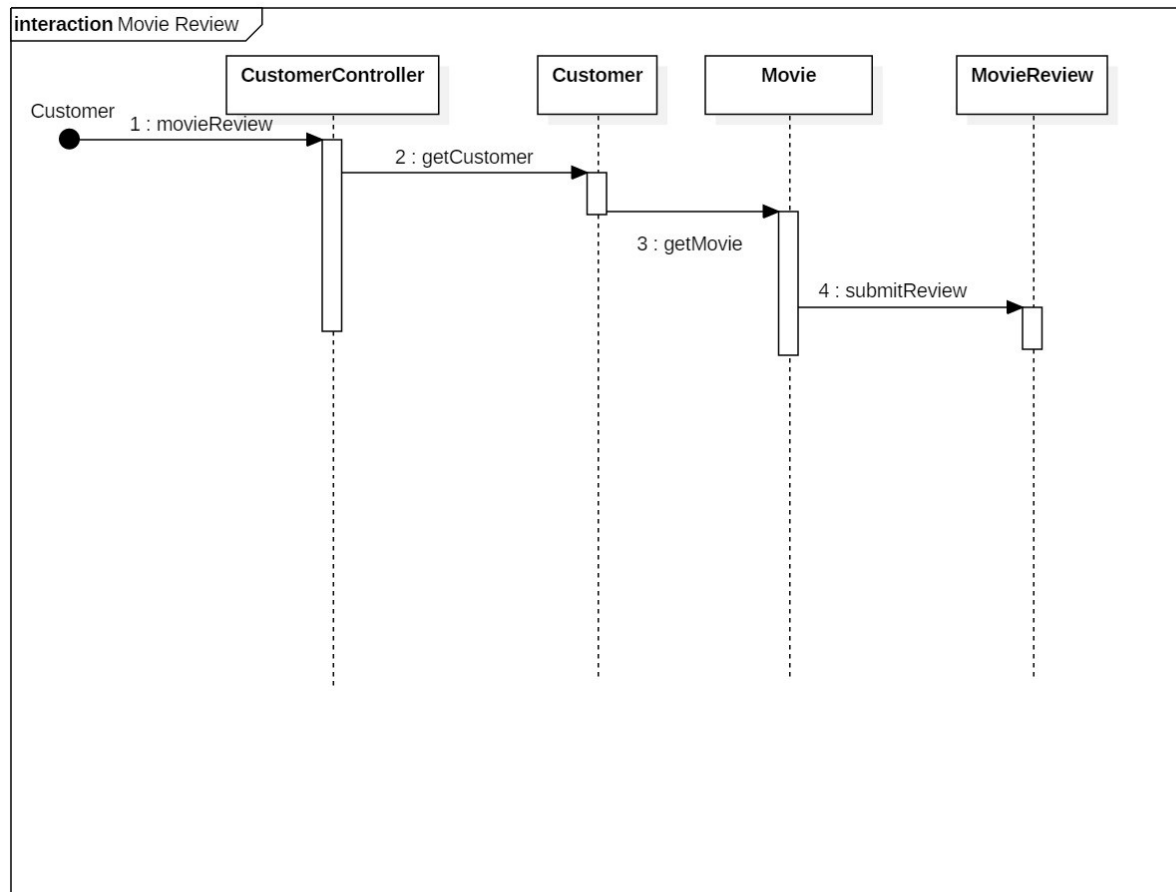3 : getMovie
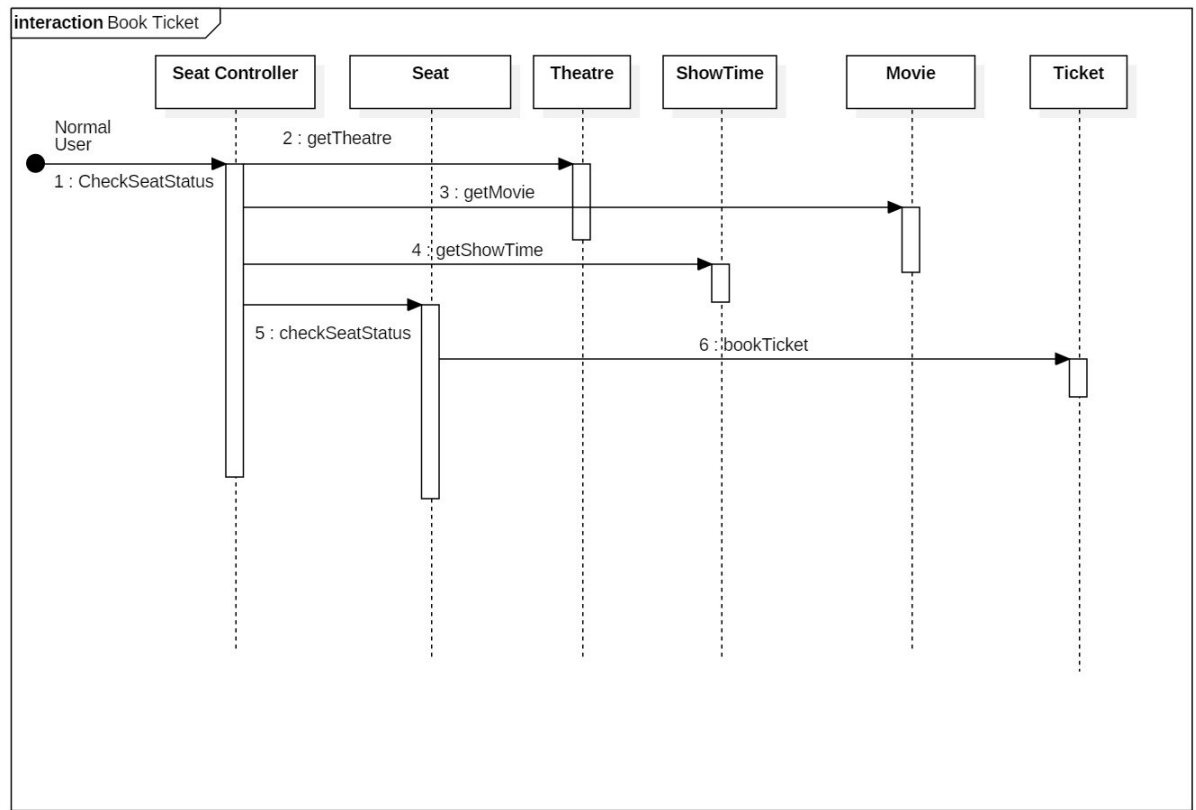
4 : submitReview

Implementation

The project is implemented using javaFX, Scene builder, IntelliJ IDE for the front end and MySql database as back end. For the implementation purpose, most of the Object Oriented Principles are applied. All the code is written in object oriented approach and design patterns like Factory method and singleton methods are applied. Program is designed in MVC model.

Conclusion

Project Movie Ticketing System is completed by using Object Oriented paradigm of programming with major functionalities only (because of constraints of limited time frame). There are several room for enhancement and update for future work.