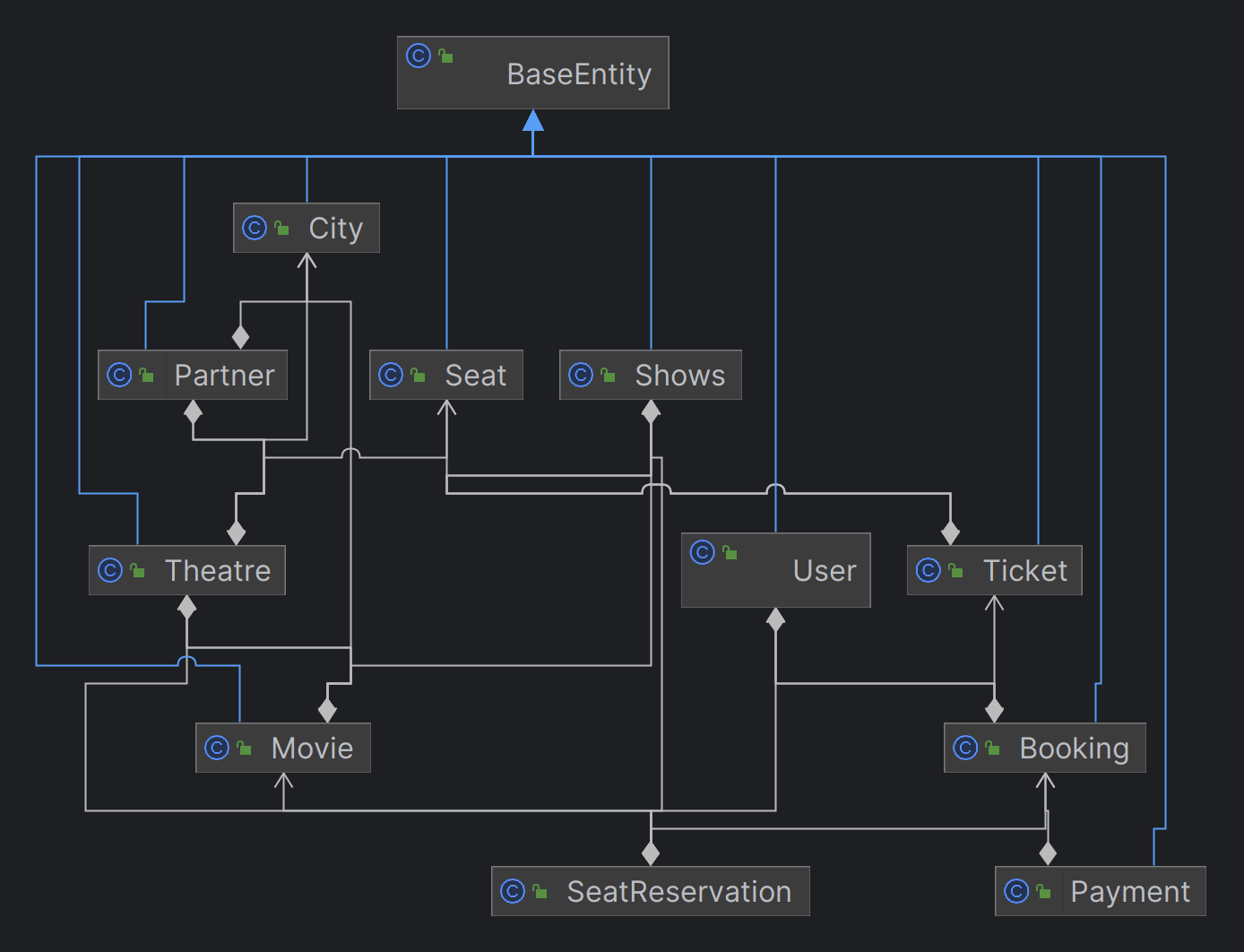
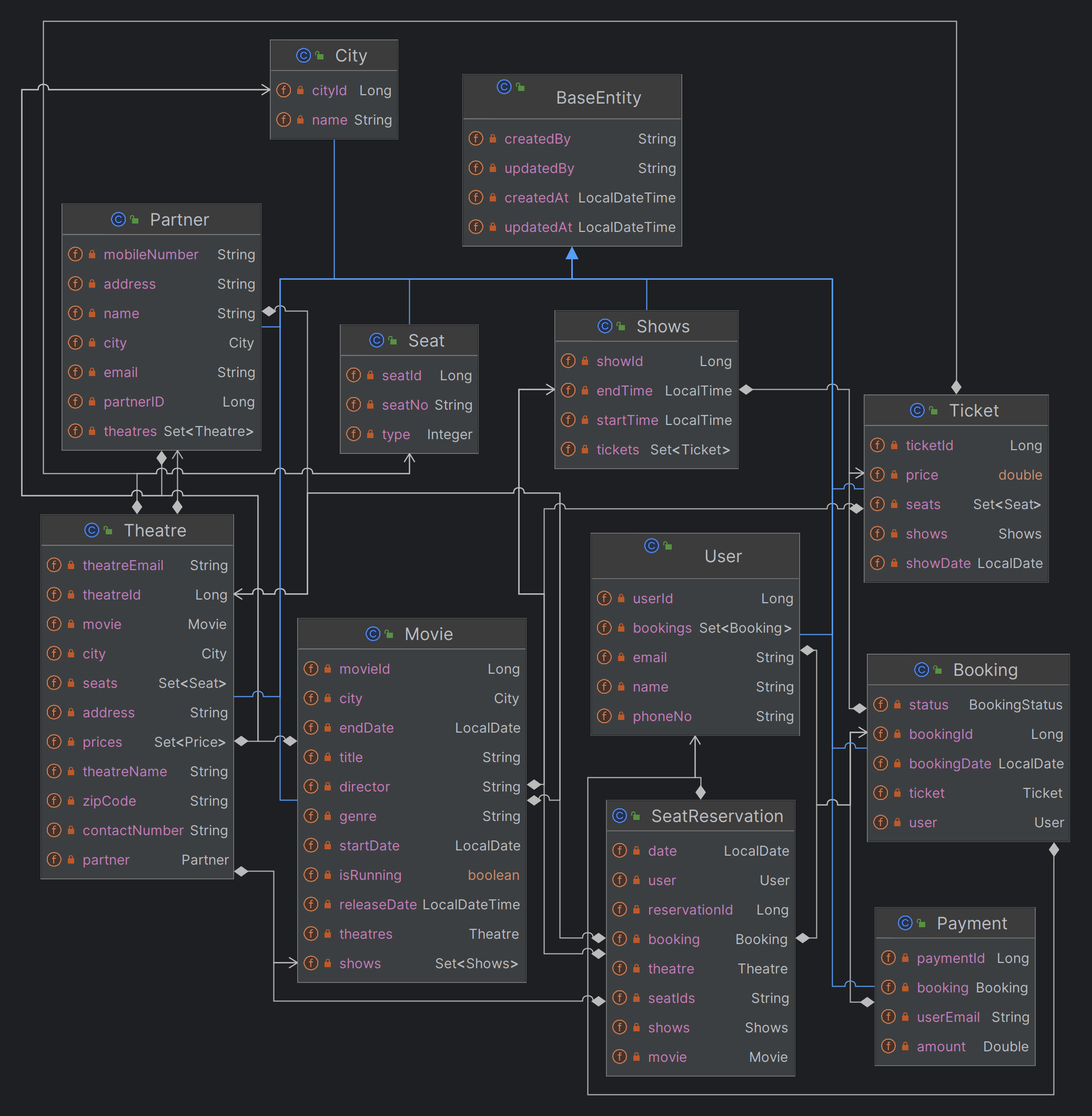
Data Model



Data Model with Attributes



Basic Platform Architecture.

APIs

**Controller**

Application Logic

Request Mapping

Services

Repositiry

Mapper

DB

DTO

Model

APIs Contract

Partner Onboarding APIs

1. Method -POST

<http://localhost:8080/bookmymovie/app/partner/onboarding>

Creates a partner that can later be associated with theatre and movies

1. Method -GET

<http://localhost:8080/bookmymovie/app/parners>

fetches the list of partners

1. METHOD- GET

[http://localhost:8080/bookmymovie/app/partners/{id}](http://localhost:8080/bookmymovie/app/partners/%7bid%7d)

fetches the partner with the corresponding id.

Theatre Management APIs

1. Method - POST

<http://localhost:8080/bookmymovie/app/theatre>

Creates a Theatre

1. Method- GET

<http://localhost:8080/bookmymovie/app/theatres>

fetches the list of all theatres

1. Method -GET

[http://localhost:8080/bookmymovie/app/theatre/{id}](http://localhost:8080/bookmymovie/app/theatre/%7bid%7d)

fetches the details of theratre with corresponding id

Movie Management APIs

1. Method - POST

<http://localhost:8080/bookmymovie/app/movie>

Creates a Movie in a theatre with all the shows

1. Method- GET

<http://localhost:8080/bookmymovie/app/movies>

fetches the list of all Movies

1. Method -GET

[http://localhost:8080/bookmymovie/app/movies/{id}](http://localhost:8080/bookmymovie/app/movies/%7bid%7d)

fetches the details of movie with corresponding id

1. Method- GET

<http://localhost:8080/bookmymovie/app/movies/search>

Searches the movies based on movie name,date and city name passes as path parameters

Booking Management APIs

1. Method - POST

<http://localhost:8080/bookmymovie/app/booking>

Books a movie based on movieId,theatreId,,showId,userID,list of seatsId and booking date

1. Method- POST

<http://localhost:8080/bookmymovie/app/booking/ticket>

creates ticket for the booking

1. Method -GET

[http://localhost:8080/bookmymovie/app/booking/status/{id}](http://localhost:8080/bookmymovie/app/booking/status/%7bid%7d)

fetches the status of the corresponding bookingid

Functional Features

Read Scenario

**Anyone of the following read scenarios:** (Only Service Implementation needed/ No UI required)

* Browse theatres currently running the show (movie selected) in the town, including show timing by a chosen date

API for browsing the theatres currently running the show.

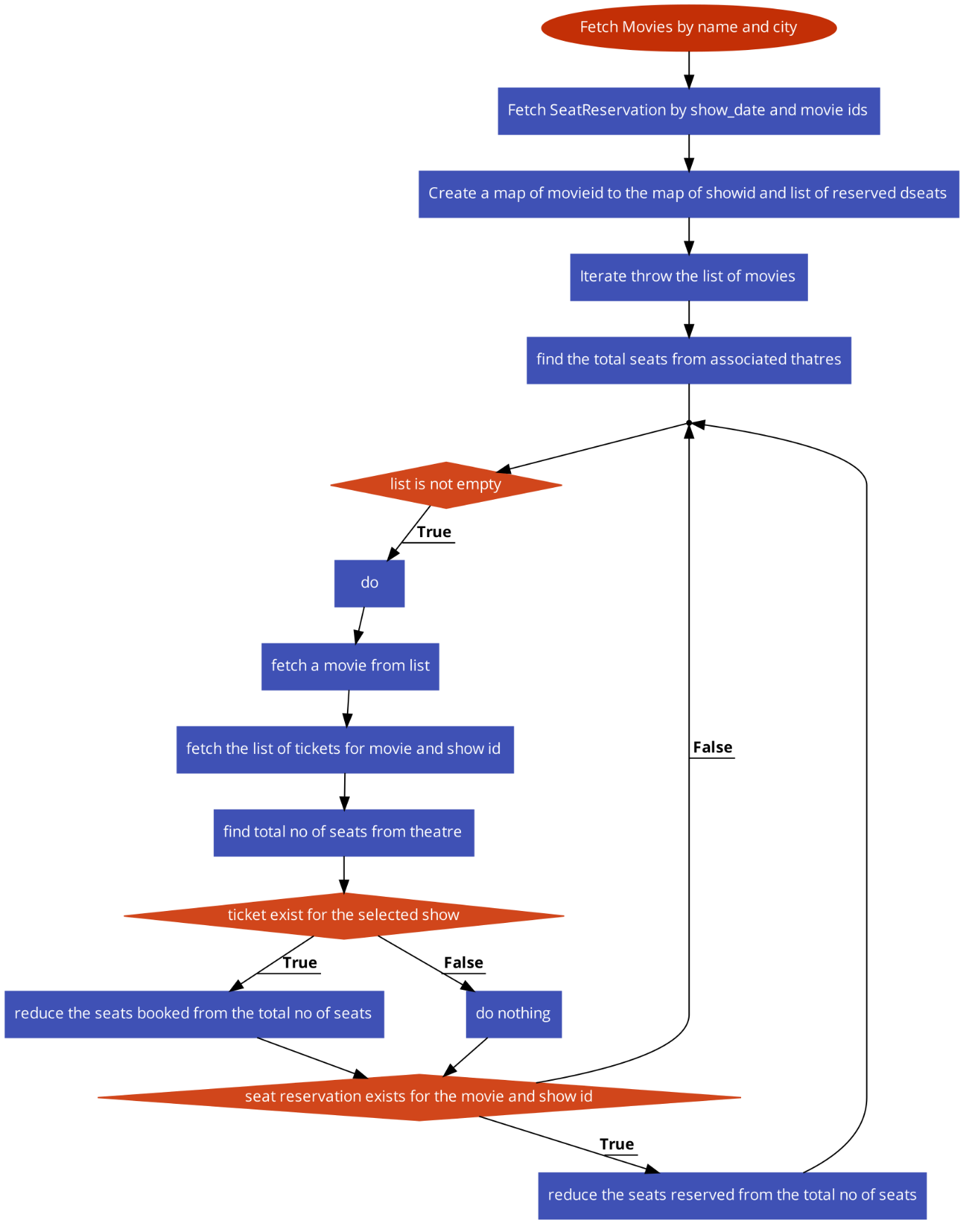
/bookmymovie/api/movies/search?name=’incep’&city=’noida’ & date=’2023-12-01’

Sequence Diagram :

A diagram of a project

Description automatically generated with medium confidence

Flow diagram of logic:



**Anyone of the following write scenarios:** Good to have - Code Implementation (write scenario):

* Book movie tickets by selecting a theatre, timing, and preferred seats for the day

Booking of Movie ticket happens in multi steps

1. Client calls the booking api
2. Booking api resrves the seats, generates an event in RabbitMQ for notification to the payment service and immediately returns the response to the client.
3. On other side notification subscriber consumes the event and calls the payment service.
4. Payment service process the payments and publishes an event for the booking consumer.
5. Booking consumer consumes the event and calls the booking service .
6. Booking service creates a ticket if the payment is successful.
7. If the payment is not successful then updates the booking status as failed .

Sequence diagram:

A diagram of a movie ticket

Description automatically generated

Sequence diagram of BookingController

A diagram of a application

Description automatically generated

Sequence diagram of Payment api

A diagram of a payment process

Description automatically generated