ONLINE MOVIE TICKET BOOKING SYSTEM

Project Description: The Online Movie Ticket Booking System is a web-based tool that is used to help in managing the operations in a multiplex as per our needs and can manage all the operations like booking ticket, managing cities, managing multiplexes, managing theatres, managing users, and managing admins.

Project Objectives: The goal of the Online Movie Ticket System project is to create an organic and optimal way of managing all the required operations in a Movie theatre. The Project mainly helps on maintaining the records of every aspect which is easy to manage the Multiplex.

Project Scope: Depending on the project performance, Super Admin, Admin and Customers can use the web-based system for an extent purpose like managing, creating, adding theatres, bookings etc.,

- Super Admin Role
    - Login.
    - Add, Update, Delete, Search, View Admins.
    - Delete, Search, View Bookings.
    - Change Password
    - Forgot/Verify Password
    - Logout.

- Admin Role
    - Login
    - Add, Update, Delete, Search, View Cities.
    - Add, Update, Delete, Search, View Location.
    - Add, Update, Delete, Search, View Movie.
    - Add, Update, Delete, Search, View Multiplex.
    - Add, Update, Delete, Search, View Theatre.
    - Update, Delete, Search, View Bookings.
    - Add, Update, Delete, Search, View Customers.
    - View, Update Profile.
    - Add Balance.
    - Checking, Booking and Cancelling Tickets.
    - Logout.

- Customer Role
    - Login.
    - View Cities.
    - View, Search Movies.
    - View Theatres.
    - Checking, Booking and Cancelling Tickets.
    - View, Update Profile.
    - Add Balance.
    - Forgot Password.
    - Logout.

- Guest Role
    - Register.
    - View Cities.
    - View, Search Movies.
    - View Theatres.
    - Checking Available Seats.

Online Movie Ticket Booking System DB Schema, Use Case Diagram and Class Diagrams

The ER diagram for the Online Movie Ticket Booking System was made based on some requirements. It can Manage Admins, Customers, Cities, Locations, Multiplexes, Theatres, Bookings etc.

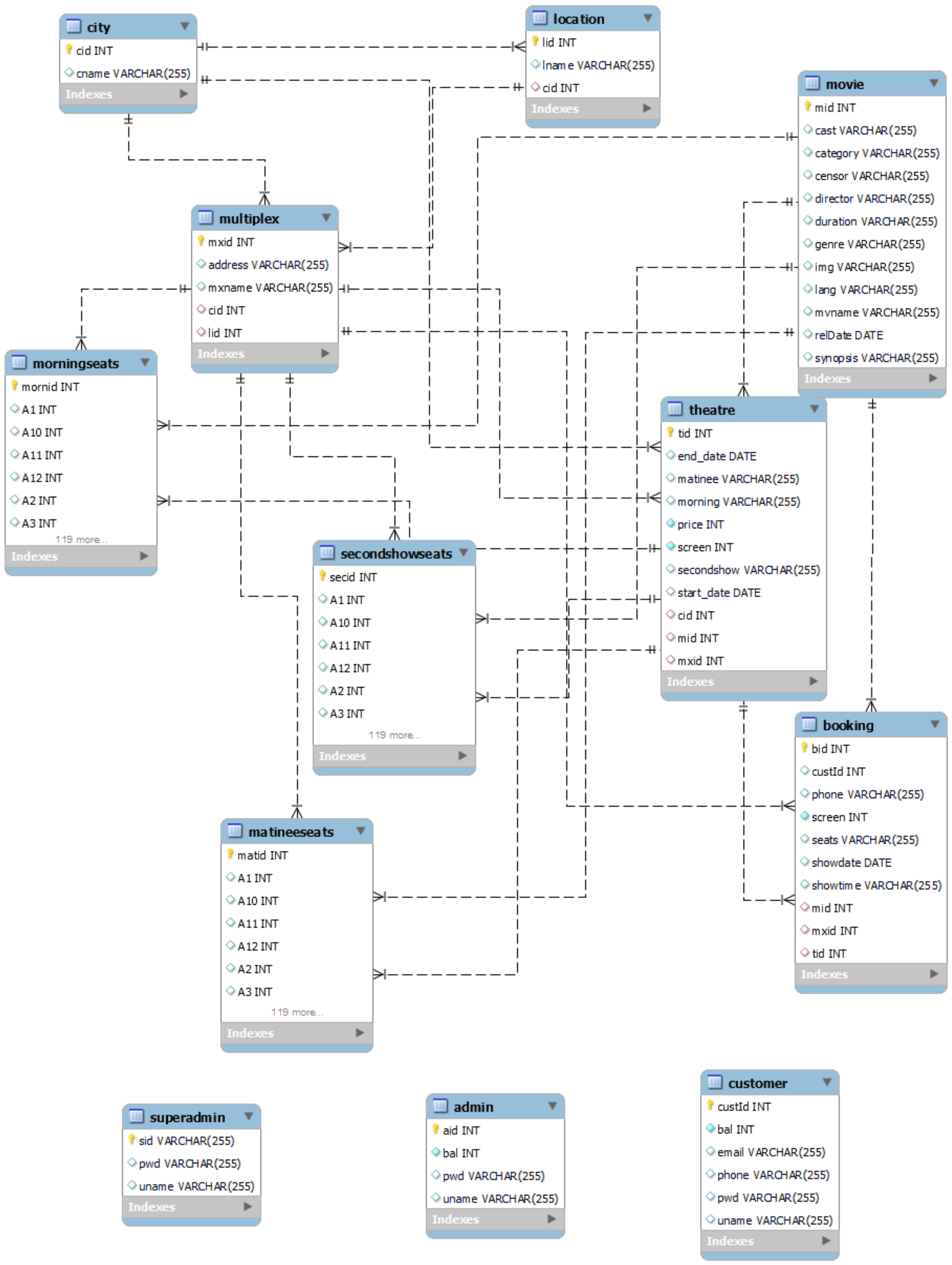
Admin can have access to manage Bookings, manage Cities, manage Locations, Manage Locations, manage Movies, manage Multiplex, manage Theatre, and changing his password. But he doesn't have the access to view or change any other Admin.

Super Admin’s only purpose is to manage Admins and view Bookings.

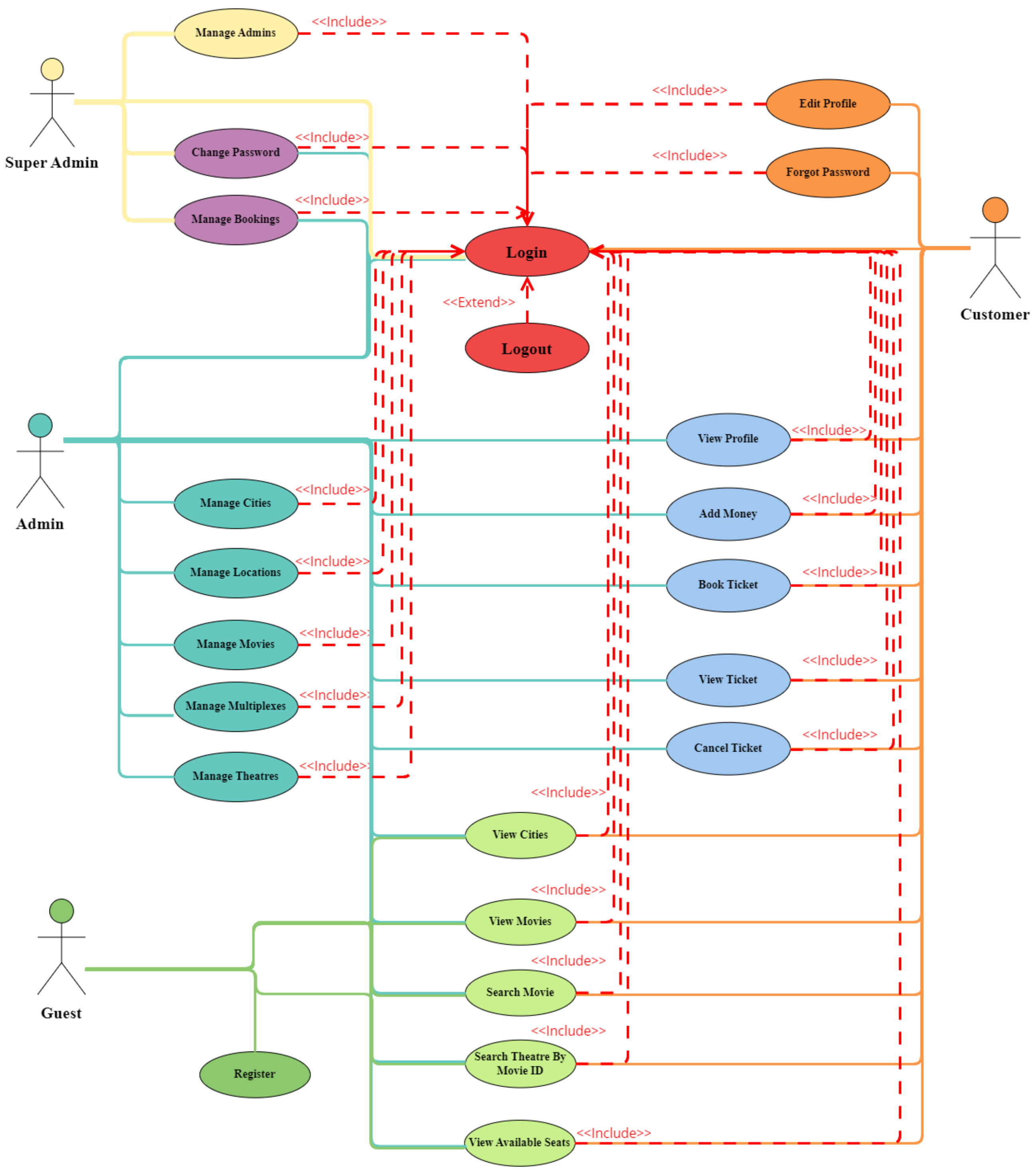
Customer Can only view, Book Movie, and edit his Details.

Guest can only view the Movies and Register.

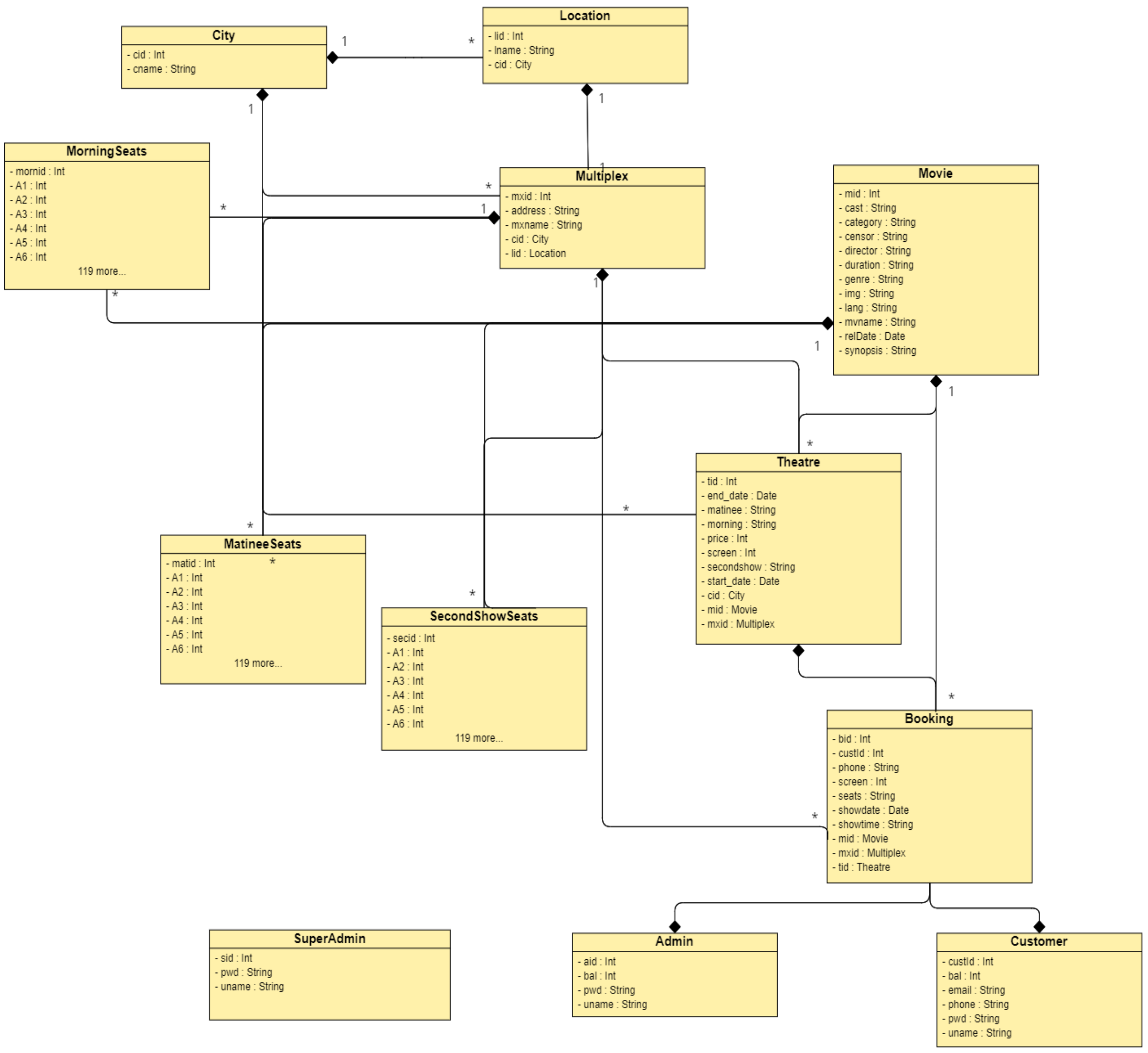
Database Schema:



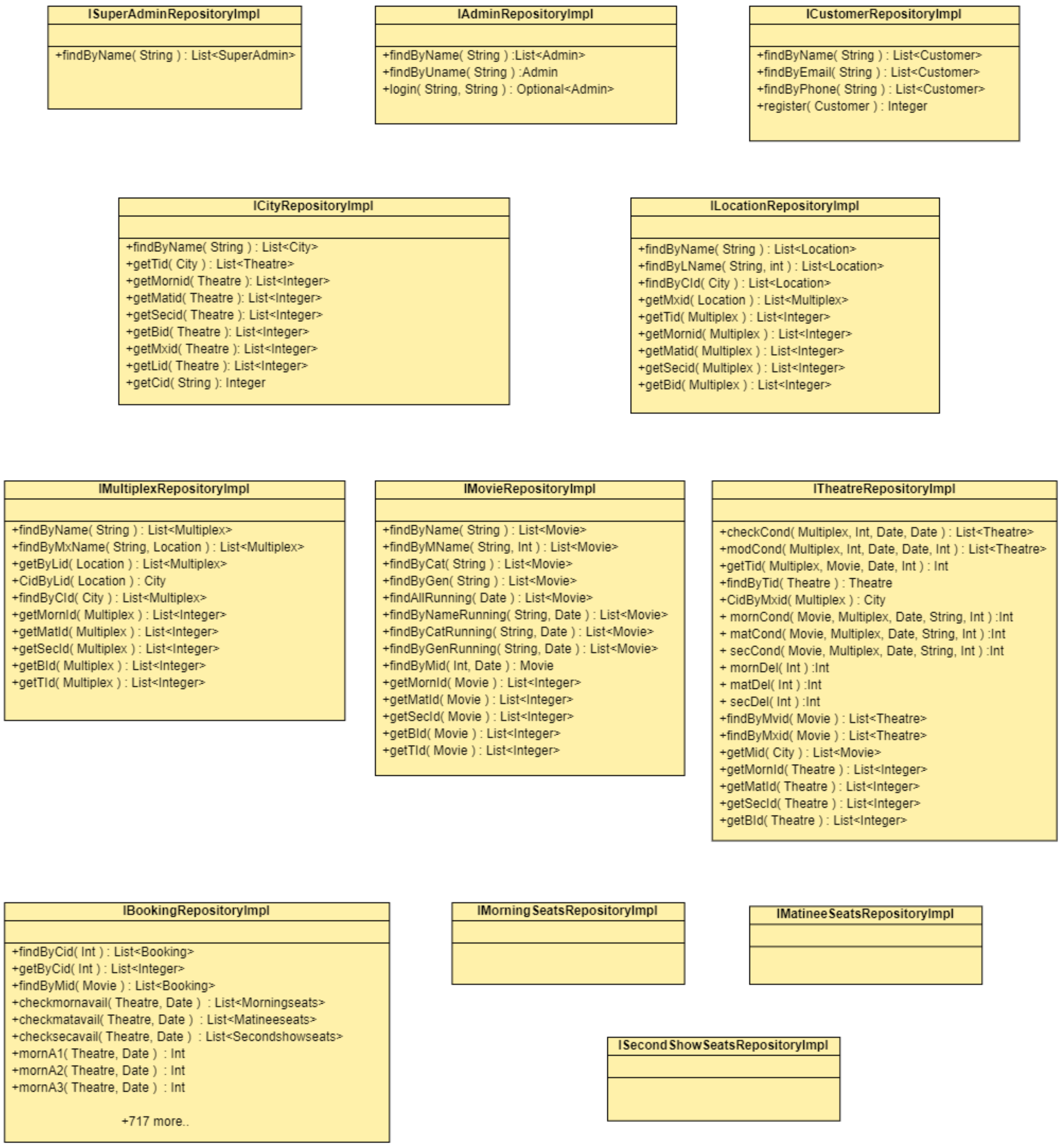
Use Case Diagram:



Class Diagram(Pojo):



Class Diagram(Repository):



```
ISuperAdminRepositoryImpl
+findByName( String ) : List<SuperAdmin>
```

```
IAdminRepositoryImpl
+findByName( String ) :List<Admin>
+findByUname( String ) :Admin
+login( String, String ) : Optional<Admin>
```

```
ICustomerRepositoryImpl
+findByName( String ) : List<Customer>
+findByEmail( String ) : List<Customer>
+findByPhone( String ) : List<Customer>
+register( Customer ) : Integer
```

```
ICityRepositoryImpl
+findByName( String ) : List<City>
+getTid( City ) : List<Theatre>
+getMornid( Theatre ): List<Integer>
+getMatid( Theatre ): List<Integer>
+getSecid( Theatre ): List<Integer>
+getBid( Theatre ): List<Integer>
+getMxid( Theatre ): List<Integer>
+getLid( Theatre ): List<Integer>
+getCid( String ): Integer
```

```
ILocationRepositoryImpl
+findByName( String ) : List<Location>
+findByLName( String, int ) : List<Location>
+findByCId( City ) : List<Location>
+getMxid( Location ) : List<Multiplex>
+getTid( Multiplex ) : List<Integer>
+getMornid( Multiplex ) : List<Integer>
+getMatid( Multiplex ) : List<Integer>
+getSecid( Multiplex ) : List<Integer>
+getBid( Multiplex ) : List<Integer>
```

```
IMultiplexRepositoryImpl
+findByName( String ) : List<Multiplex>
+findByMxName( String, Location ) : List<Multiplex>
+getByLid( Location ) : List<Multiplex>
+CidByLid( Location ) : City
+findByCId( City ) : List<Multiplex>
+getMornId( Multiplex ) : List<Integer>
+getMatId( Multiplex ) : List<Integer>
+getSecId( Multiplex ) : List<Integer>
+getBId( Multiplex ) : List<Integer>
+getTId( Multiplex ) : List<Integer>
```

```
IMovieRepositoryImpl
+findByName( String ) : List<Movie>
+findByMName( String, Int ) : List<Movie>
+findByCat( String ) : List<Movie>
+findByGen( String ) : List<Movie>
+findAllRunning( Date ) : List<Movie>
+findByNameRunning( String, Date ) : List<Movie>
+findByCatRunning( String, Date ) : List<Movie>
+findByGenRunning( String, Date ) : List<Movie>
+findByMid( Int, Date ) : Movie
+getMornId( Movie ) : List<Integer>
+getMatId( Movie ) : List<Integer>
+getSecId( Movie ) : List<Integer>
+getBId( Movie ) : List<Integer>
+getTId( Movie ) : List<Integer>
```

```
ITheatreRepositoryImpl
+checkCond( Multiplex, Int, Date, Date ) : List<Theatre>
+modCond( Multiplex, Int, Date, Date, Int ) : List<Theatre>
+getTid( Multiplex, Movie, Date, Int ) : Int
+findByTid( Theatre ) : Theatre
+CidByMxid( Multiplex ) : City
+ mornCond( Movie, Multiplex, Date, String, Int ) :Int
+ matCond( Movie, Multiplex, Date, String, Int ) :Int
+ secCond( Movie, Multiplex, Date, String, Int ) :Int
+ mornDel( Int ) :Int
+ matDel( Int ) :Int
+ secDel( Int ) :Int
+findByMvid( Movie ) : List<Theatre>
+findByMxid( Movie ) : List<Theatre>
+getMid( City ) : List<Movie>
+getMornId( Theatre ) : List<Integer>
+getMatId( Theatre ) : List<Integer>
+getSecId( Theatre ) : List<Integer>
+getBId( Theatre ) : List<Integer>
```

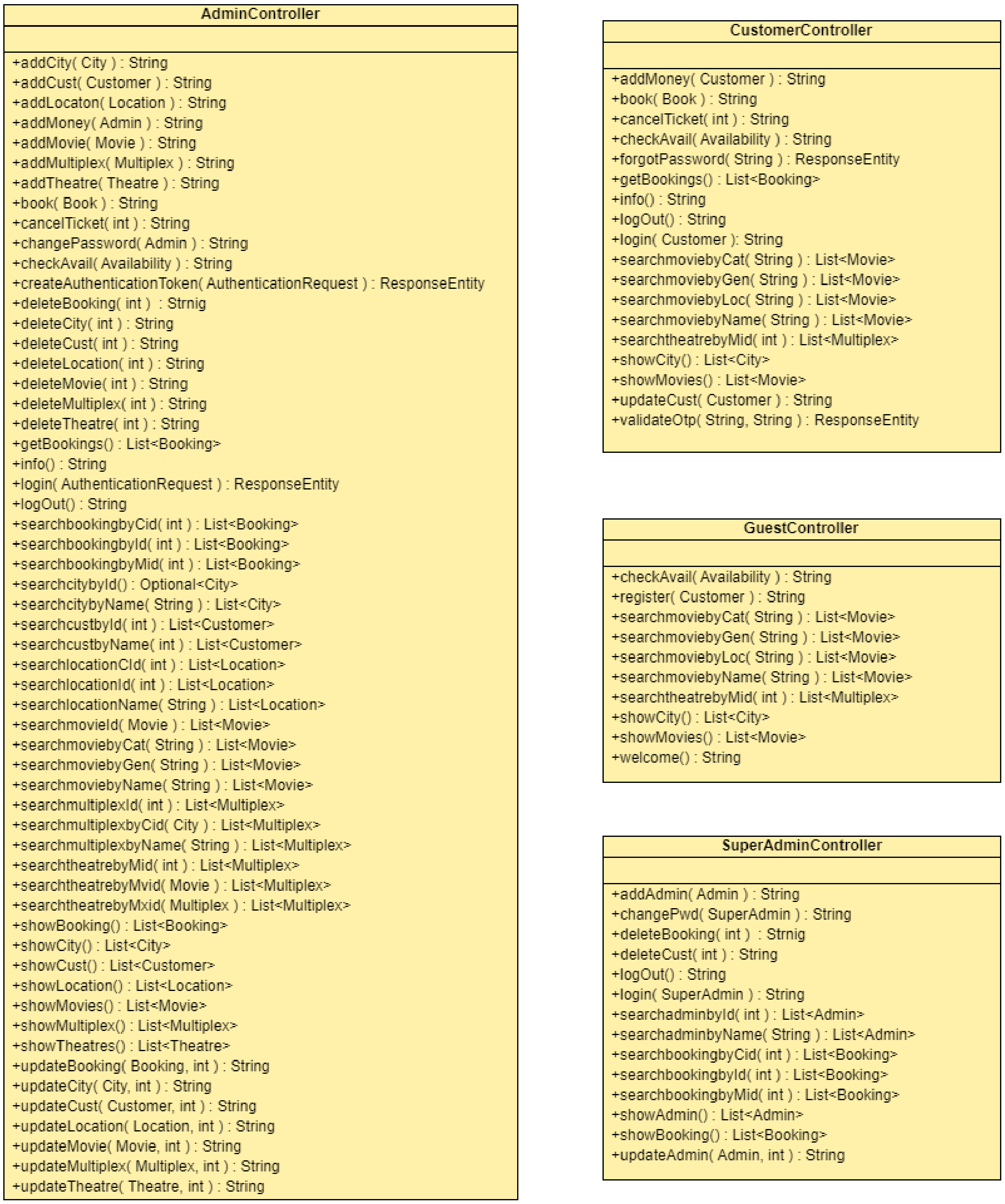
```
IBookingRepositoryImpl
+findByCid( Int ) : List<Booking>
+getByCid( Int ) : List<Integer>
+findByMid( Movie ) : List<Booking>
+checkmornavail( Theatre, Date ) : List<Morningseats>
+checkmatavail( Theatre, Date ) : List<Matineeseats>
+checksecavail( Theatre, Date ) : List<Secondshowseats>
+mornA1( Theatre, Date ) : Int
+mornA2( Theatre, Date ) : Int
+mornA3( Theatre, Date ) : Int

+717 more..
```

```
IMorningSeatsRepositoryImpl
```

```
IMatineeSeatsRepositoryImpl
```

```
ISecondShowSeatsRepositoryImpl
```

Class Diagram(Controller):



```
AdminController
----------------------------------------
----------------------------------------
+addCity( City ) : String
+addCust( Customer ) : String
+addLocaton( Location ) : String
+addMoney( Admin ) : String
+addMovie( Movie ) : String
+addMultiplex( Multiplex ) : String
+addTheatre( Theatre ) : String
+book( Book ) : String
+cancelTicket( int ) : String
+changePassword( Admin ) : String
+checkAvail( Availability ) : String
+createAuthenticationToken( AuthenticationRequest ) : ResponseEntity
+deleteBooking( int )  : Strnig
+deleteCity( int ) : String
+deleteCust( int ) : String
+deleteLocation( int ) : String
+deleteMovie( int ) : String
+deleteMultiplex( int ) : String
+deleteTheatre( int ) : String
+getBookings() : List<Booking>
+info() : String
+login( AuthenticationRequest ) : ResponseEntity
+logOut() : String
+searchbookingbyCid( int ) : List<Booking>
+searchbookingbyId( int ) : List<Booking>
+searchbookingbyMid( int ) : List<Booking>
+searchcitybyId() : Optional<City>
+searchcitybyName( String ) : List<City>
+searchcustbyId( int ) : List<Customer>
+searchcustbyName( int ) : List<Customer>
+searchlocationCId( int ) : List<Location>
+searchlocationId( int ) : List<Location>
+searchlocationName( String ) : List<Location>
+searchmovieId( Movie ) : List<Movie>
+searchmoviebyCat( String ) : List<Movie>
+searchmoviebyGen( String ) : List<Movie>
+searchmoviebyName( String ) : List<Movie>
+searchmultiplexId( int ) : List<Multiplex>
+searchmultiplexbyCid( City ) : List<Multiplex>
+searchmultiplexbyName( String ) : List<Multiplex>
+searchtheatrebyMid( int ) : List<Multiplex>
+searchtheatrebyMvid( Movie ) : List<Multiplex>
+searchtheatrebyMxid( Multiplex ) : List<Multiplex>
+showBooking() : List<Booking>
+showCity() : List<City>
+showCust() : List<Customer>
+showLocation() : List<Location>
+showMovies() : List<Movie>
+showMultiplex() : List<Multiplex>
+showTheatres() : List<Theatre>
+updateBooking( Booking, int ) : String
+updateCity( City, int ) : String
+updateCust( Customer, int ) : String
+updateLocation( Location, int ) : String
+updateMovie( Movie, int ) : String
+updateMultiplex( Multiplex, int ) : String
+updateTheatre( Theatre, int ) : String
```

```
CustomerController
----------------------------------------
----------------------------------------
+addMoney( Customer ) : String
+book( Book ) : String
+cancelTicket( int ) : String
+checkAvail( Availability ) : String
+forgotPassword( String ) : ResponseEntity
+getBookings() : List<Booking>
+info() : String
+logOut() : String
+login( Customer ): String
+searchmoviebyCat( String ) : List<Movie>
+searchmoviebyGen( String ) : List<Movie>
+searchmoviebyLoc( String ) : List<Movie>
+searchmoviebyName( String ) : List<Movie>
+searchtheatrebyMid( int ) : List<Multiplex>
+showCity() : List<City>
+showMovies() : List<Movie>
+updateCust( Customer ) : String
+validateOtp( String, String ) : ResponseEntity
```

```
GuestController
----------------------------------------
----------------------------------------
+checkAvail( Availability ) : String
+register( Customer ) : String
+searchmoviebyCat( String ) : List<Movie>
+searchmoviebyGen( String ) : List<Movie>
+searchmoviebyLoc( String ) : List<Movie>
+searchmoviebyName( String ) : List<Movie>
+searchtheatrebyMid( int ) : List<Multiplex>
+showCity() : List<City>
+showMovies() : List<Movie>
+welcome() : String
```

```
SuperAdminController
----------------------------------------
----------------------------------------
+addAdmin( Admin ) : String
+changePwd( SuperAdmin ) : String
+deleteBooking( int )  : Strnig
+deleteCust( int ) : String
+logOut() : String
+login( SuperAdmin ) : String
+searchadminbyId( int ) : List<Admin>
+searchadminbyName( String ) : List<Admin>
+searchbookingbyCid( int ) : List<Booking>
+searchbookingbyId( int ) : List<Booking>
+searchbookingbyMid( int ) : List<Booking>
+showAdmin() : List<Admin>
+showBooking() : List<Booking>
+updateAdmin( Admin, int ) : String
```

Functional Requirements

1. Super Admin
    a. Super Admin Can Manage Admins.
    b. Super Admin Can Manage Bookings.
    c. Super Admin Can Change his Password.

2. Admin
    a. Admin Can Manage Cities.
    b. Admin Can Manage Locations.
    c. Admin Can Manage Movies.
    d. Admin Can Manage Multiplex.
    e. Admin Can Manage Theatres.
    f. Admin Can Manage Bookings
    g. Admin Can Manage Customers.
    h. Admin Can Change his Password.

3. Customer
    a. Customer Can View and Search Movies
    b. Customer Can View Theatres.
    c. Customer Can Manage his Bookings.
    d. Customer Can Mange his Profile

3. Guest

    a. Guest Can View and Search Movies
    b. Guest Can View Theatres.
    c. Guest Can Register.

Technical Requirements

- Database – MYSQL
- Backend – Spring Boot, JPA, Hibernate
- Language – Java 8
- Security– JWT
- Postman as a Frontend to perform operations.