



PES University, Bengaluru

(Established under Karnataka Act 16 of 2013)

Department of Computer Science & Engineering

Session: Jan - May 2022

UE19CS353 – Object Oriented Analysis and Design with Java

Theory ISA (Mini Project)

Report on

Movie Ticket Booking System

By:

Name	SRN
Tankala Sunaina	PES2UG19CS427
Toshani Rungta	PES2UG19CS4333
Venkata Krishnarjun Vuppala	PES2UG19CS451

6th Semester G

1. Project Description

a. Link to Github repository

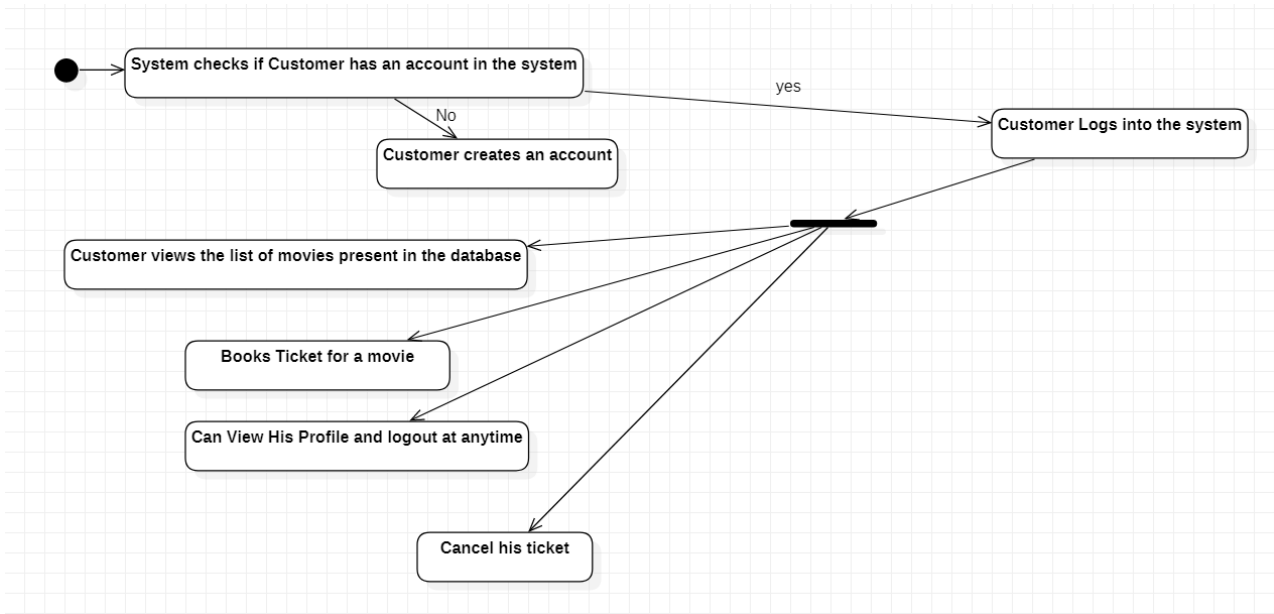
<https://github.com/theprojectrepo123/Movie-Ticket-Booking-System>

2. Analysis and Design Models

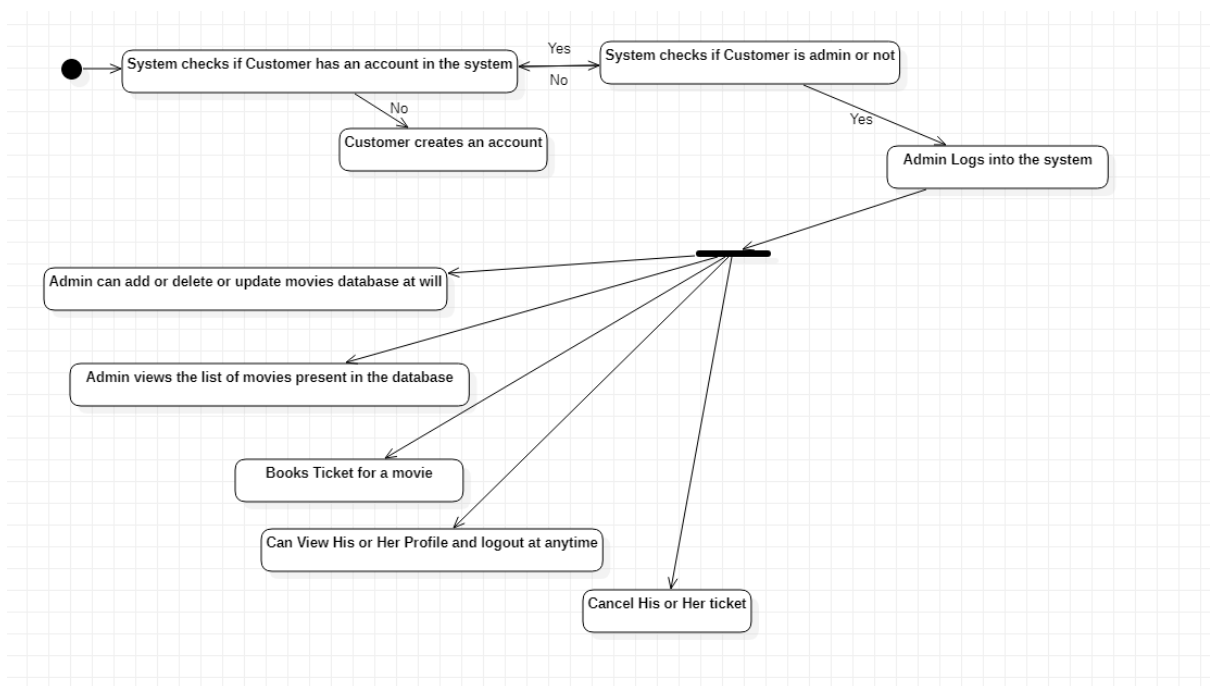
- Use Case Diagram:



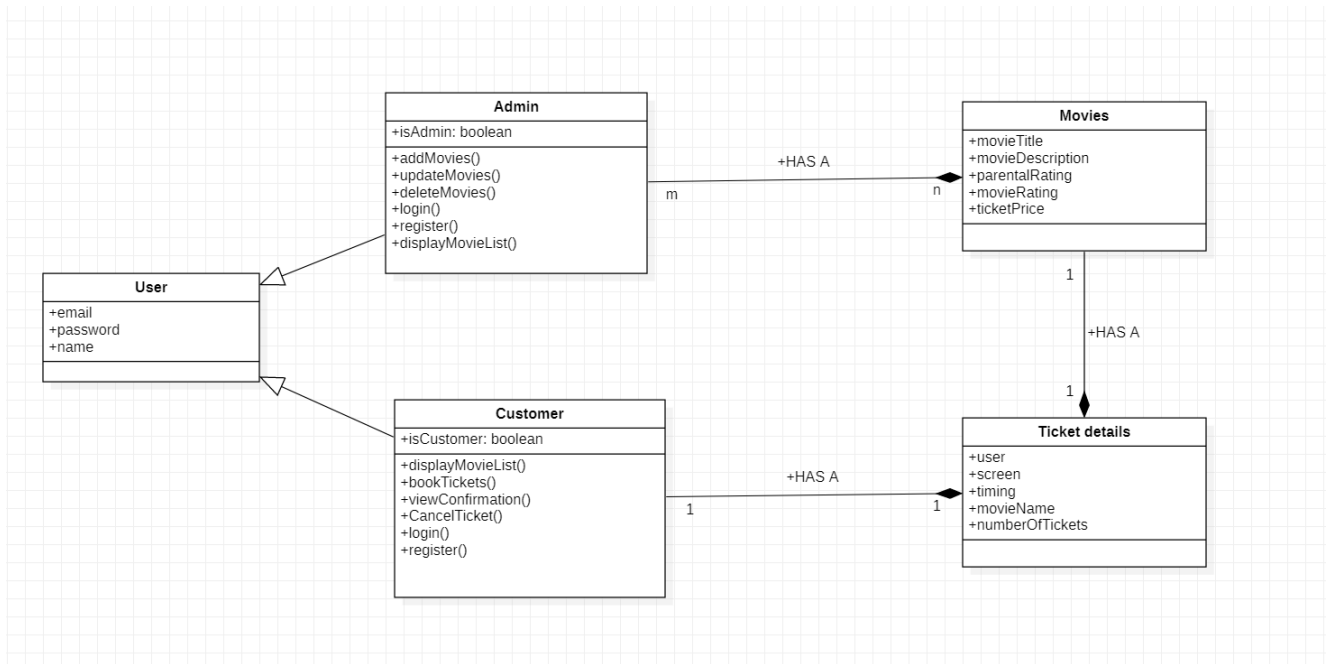
- **State Transition Diagram:**
 - **For Customer:**



- **State Transition Diagram:**
 - **For Admin:**



- **Class Diagram:**



3. Tools and Frameworks Used

- a. Spring Boot
- b. Hibernate
- c. Database Used: MySQL
- d. Thymeleaf
- e. Spring Devtools
- f. Spring-Security
- g. HTML/CSS/JS
- h. Bootstrap

4. Design Principles and Design Patterns Applied

- Chain Of Responsibilities (Behavioural Pattern)

This has been used in the AppController class where based on the request, it transfers control to the appropriate route handler method and if request is not valid, it redirects to an error page.

- Proxy Pattern (Structural Pattern)

The User doesn't interact directly with the model, he/she interacts with the view, the view receives the request and sends it

to the controller where the controller handles the business logic expected to be done.

- Model View Controller Pattern

The Model class, View Class and Controller class are clearly and distinctively declared in the implementation of the application with the help of SpringBoot, Thymeleaf(html) and MySQL.

- Bridge Pattern (Behavioural Pattern)

There are three separate classes, service classes, etc for movies entity, movies ticket entity and User entity. Each of the respective business logic is handled separately with respect to the entity used.

- Factory Pattern (Creational Pattern)

When the user logs in, the ApplicationController class send the request to the webconfig class where that class handles the authentication of user so that registered users can login into the system.

- Singleton Pattern (Creational Pattern)

When a user is created in the database, only one instance is created. No multiple instances are created.

- Command Pattern (Behavioural Pattern)

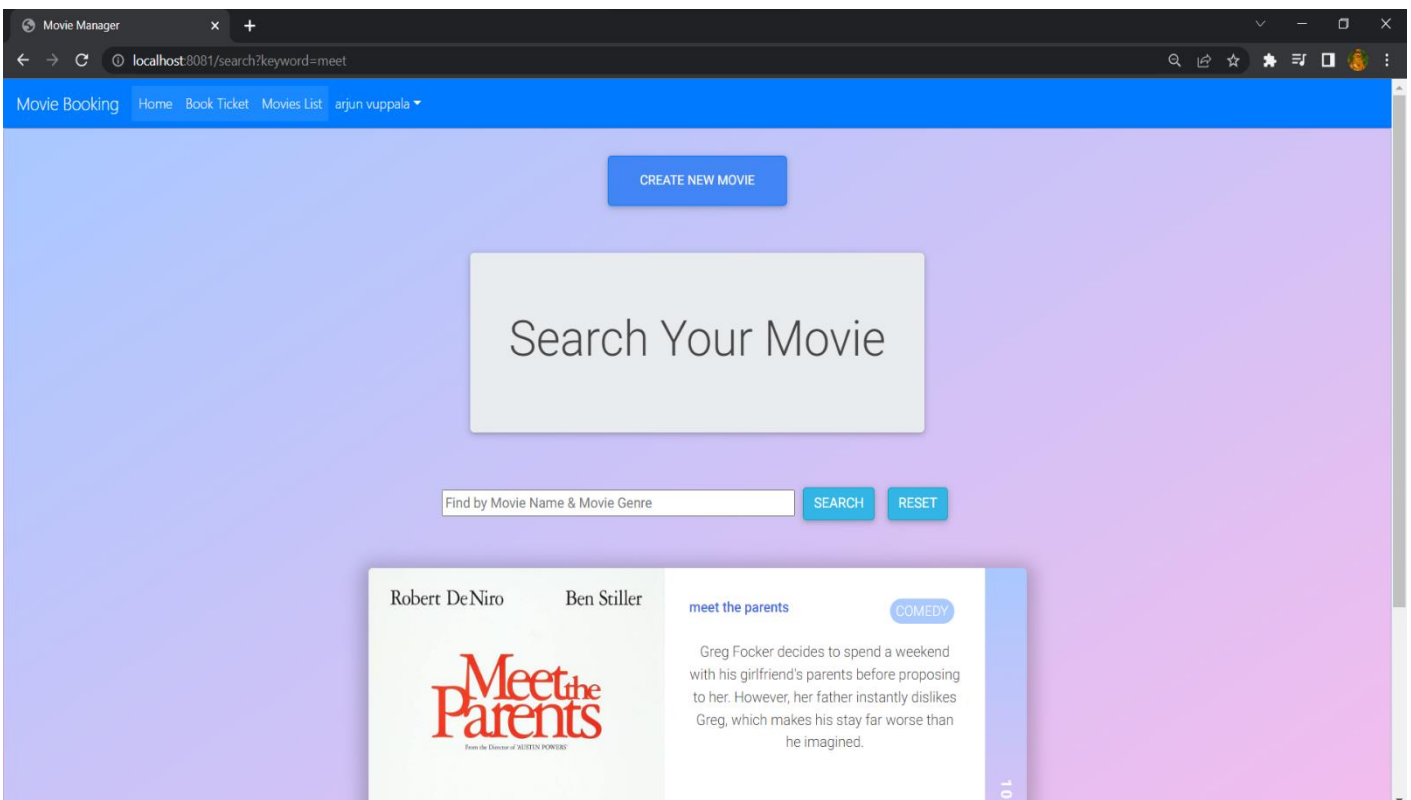
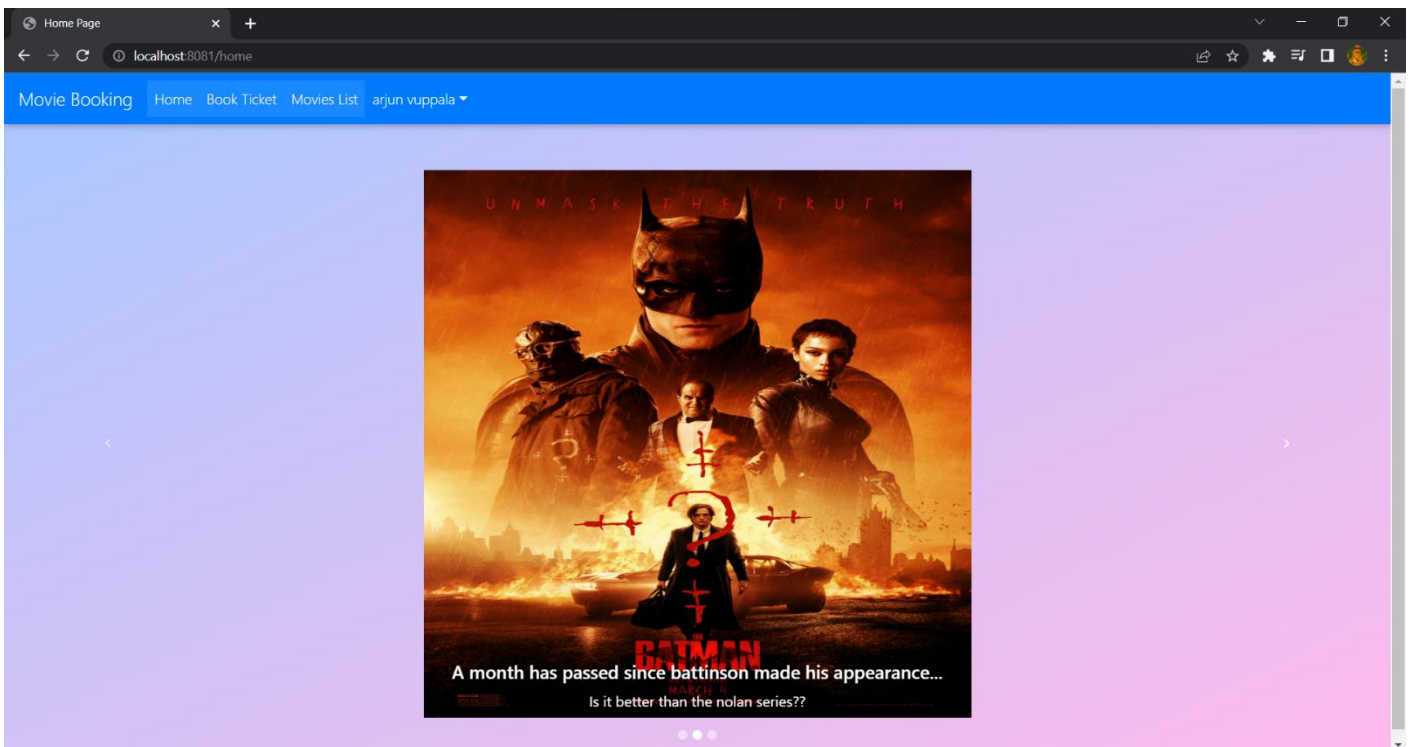
The functions for insertion, deletion and updating of the movies as well as insertion of movie tickets and users are defined in separate, distinct classes, where depending upon usage, the classes are called appropriately.

- Iterator Pattern (Behavioural Pattern)

To store the list of movies for rendering it in the movies-list page and as a dropdown in the book-ticket page, iterator pattern is used.

- Follows all SOLID principles.

5. Application Screenshots (3-4 important pages)



Registration x +

localhost:8081/register

User Registration - Sign Up

E-mail:

Password:

First Name:

Last Name:

Create New Movie x +

localhost:8081/book

Movie Booking Home Book Ticket Movies List arjun vuppala

Book Movie Ticket

Movie Name
meet the parents

User Name
arjunv2001@gmail.com

Number Of Tickets
0

Timing
9:00 am

Screen
A1

6. Team member contributions

Name	Contributions
Tankala Sunaina	Login and Register Page
Toshani Rungta	Book ticket and confirmation Page
Venkata Krishnarjun Vuppala	CRUD operations for movies list, Search in database, integration of all modules, Cancellation of Movie Ticket