



**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

**Cinema Ticket Booking System**

**By:**

**Rohith M                      PES2UG19CS336**

**Roshan Sriram            PES2UG19CS338**

**S Kalyan                    PES2UG19CS341**

**6<sup>th</sup> Semester 'F'**

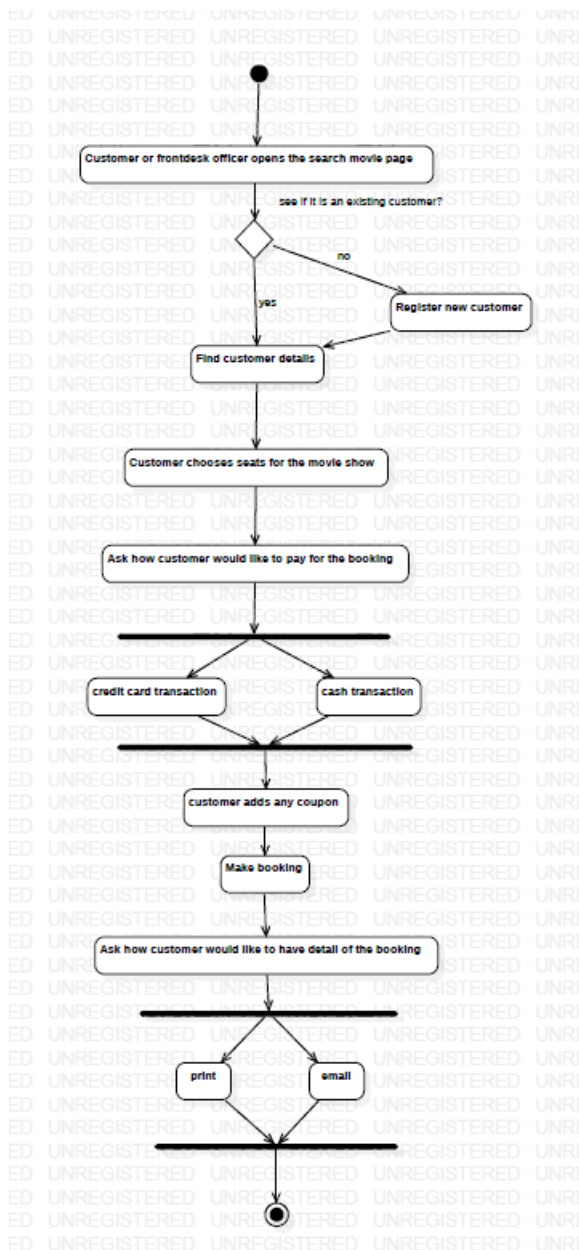
## 1. Project Description

The main purpose of our online ticket booking system is to provide an alternate and convenient way for a customer to buy cinema tickets. It is an automatic system. After the data has been fed into the database, the staff does not need to do anything with the order once it is received through the system.

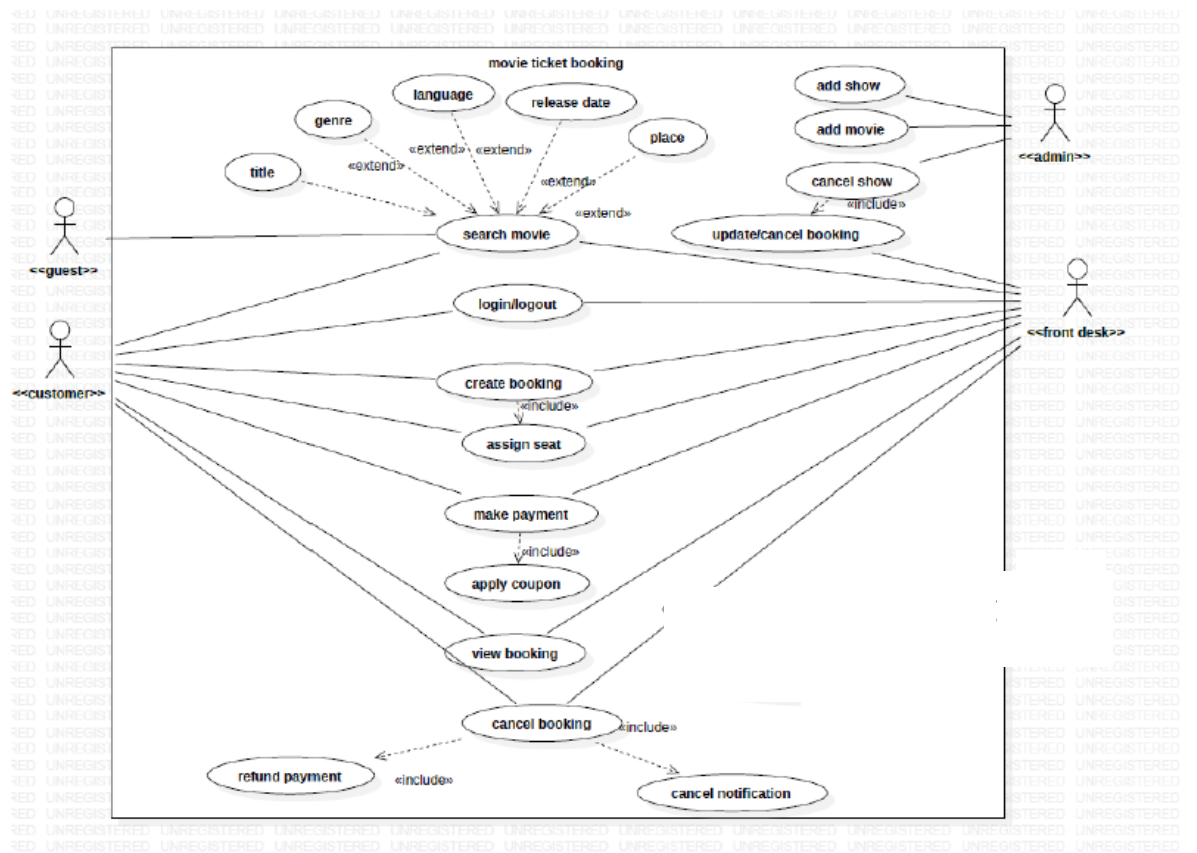
<https://github.com/rohithmgowda/movieticketbooking.git>

## 2. Analysis and Design Models

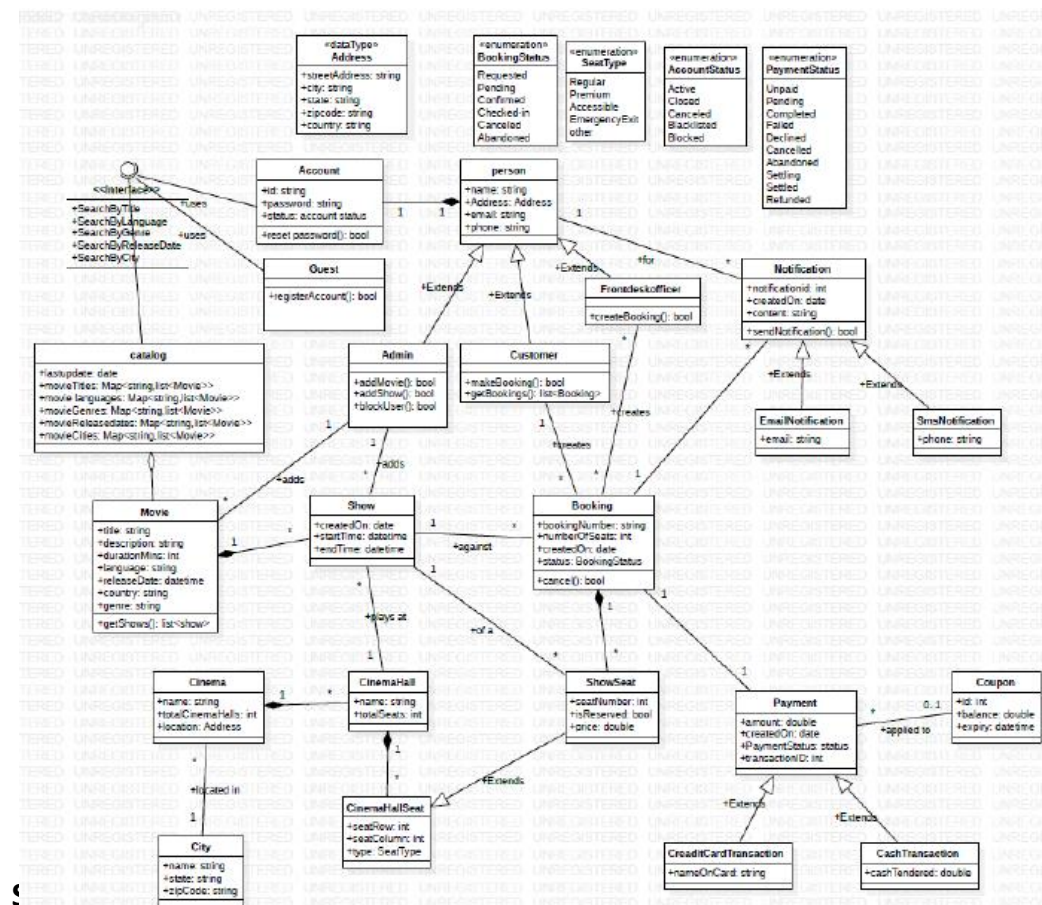
- Activity Diagram

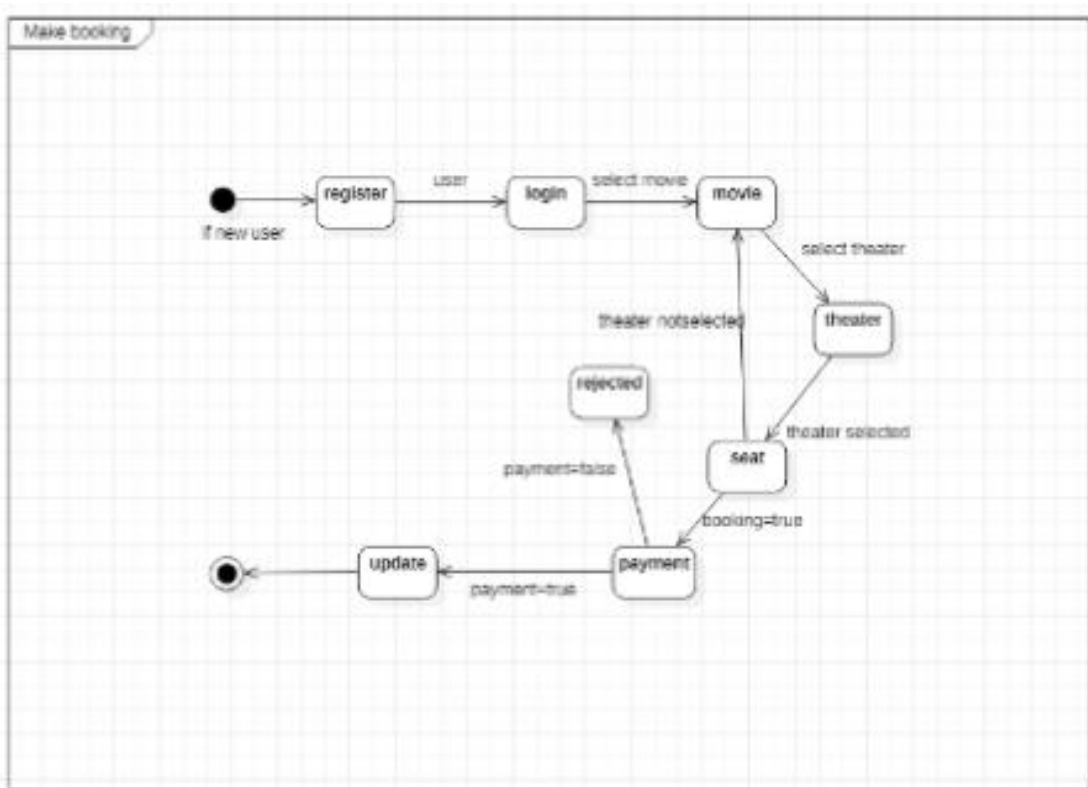


- Use Case Diagram



## ● Class Diagram





### 3. Tools and Frameworks Used

- Front-end: Netbeans-IDE
- Database: MySQL database
- Version Control: Git
- Class, activity and state diagrams: StarUML
- JDBC (Java Database Connectivity)

## 4. Design Principles and Design Patterns Applied

### Architectural patterns used

- **Layered Pattern :**

As the name suggests, components(code) in this pattern are separated into layers of subtasks and they are arranged one above another.

Each layer has unique tasks to do and all the layers are independent of one another. Since each layer is independent, one can modify the code inside a layer without affecting others.

It is the most commonly used pattern for designing the majority of software. This layer is also known as 'N-tier architecture'. Basically, this pattern has 4 layers.

1.Presentation layer (The user interface layer where we see and enter data into an application.)

2.Business layer (this layer is responsible for executing business logic as per the request.)

3.Application layer (this layer acts as a medium for communication between the 'presentation layer' and 'data layer'.

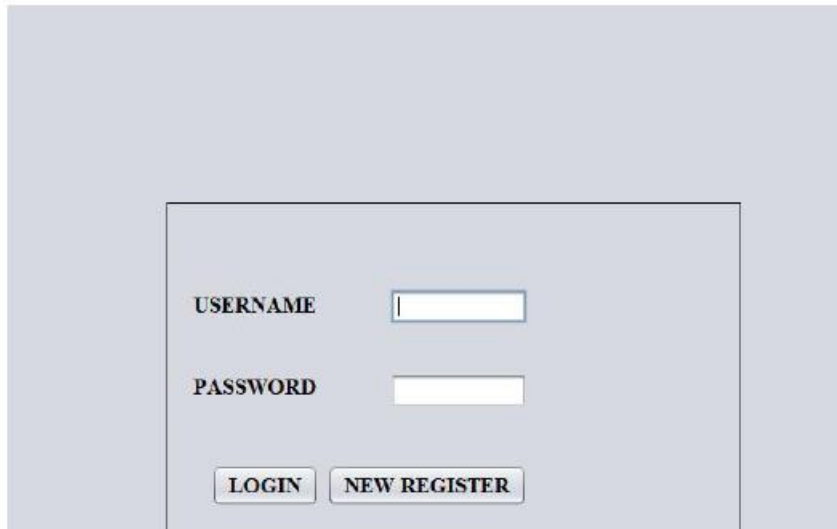
4.Data layer (this layer has a database for managing data.)

### Design pattern used:

Facade Structural Design Pattern for abstraction of complex Operations done with data. When there is even a small query done by the user such as login, the login data is collected and passed onto the model and the model searches with the database for a match and then takes an appropriate action based on this. From the user perspective the user has just given the login detail and pressed a button, but there is a lot of methods called in a chain for a suitable action to be taken.

## 5. Application Screenshots (3-4 important pages)

### 1. User login module

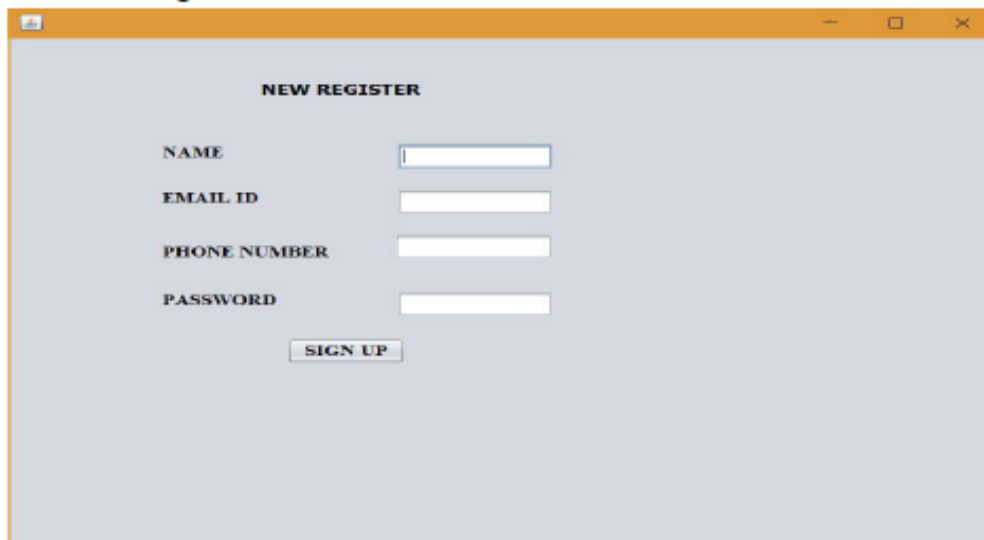


A screenshot of a user login module. It features a light blue background with a white rectangular form in the center. The form contains two text input fields: 'USERNAME' and 'PASSWORD'. Below these fields are two buttons: 'LOGIN' and 'NEW REGISTER'.

USERNAME

PASSWORD

### 2 .new registration module:



A screenshot of a new registration module. It is displayed within a window with an orange title bar. The form has a light blue background and contains four text input fields: 'NAME', 'EMAIL ID', 'PHONE NUMBER', and 'PASSWORD'. Below these fields is a 'SIGN UP' button.

**NEW REGISTER**

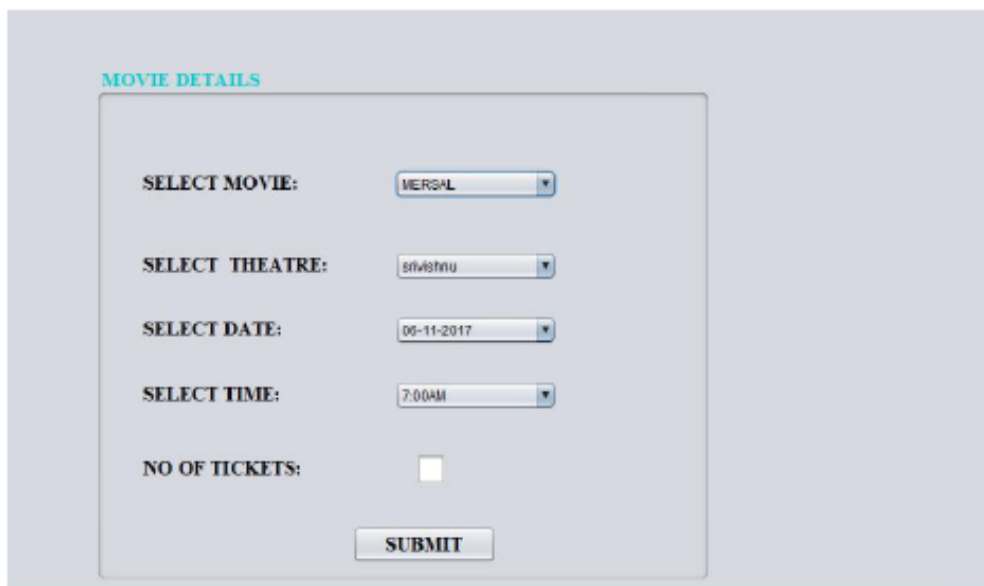
NAME

EMAIL ID

PHONE NUMBER

PASSWORD

### 3. Selection of Movie time, date, theater and ticket details:



A screenshot of a selection module for movie details. It features a light blue background with a white rectangular form in the center. The form contains five dropdown menus: 'SELECT MOVIE:', 'SELECT THEATRE:', 'SELECT DATE:', and 'SELECT TIME:'. Below these is a text input field for 'NO OF TICKETS:'. At the bottom of the form is a 'SUBMIT' button.

**MOVIE DETAILS**

SELECT MOVIE:

SELECT THEATRE:

SELECT DATE:

SELECT TIME:

NO OF TICKETS:

#### 4. Card selections:

**CONFIRM DETAILS**

<b>NO OF TICKETS:</b> 2	<b>THEATER:</b> srivishnu
<b>MOVIE:</b> MERSAL	<b>DATE:</b> 08-11-2017
<b>FARE:</b> 200	<b>TIME:</b> 7:00AM

**PAYMENT TYPE**

☐ Credit card

☒ Debit card

☐ NET BANKING

☐ PAYTM WALLET

**Make Payment** **Go Back**

#### 5. Generating Tickets :

**RECEIPT**

<b>NO OF TICKETS:</b>	2
<b>THEATRE:</b>	srivishnu
<b>MOVIE:</b>	11-11-2018
<b>DATE:</b>	7:00AM
<b>SHOW:</b>	Robo 2
<b>FARE:</b>	200

**THANKS FOR BOOKING**

## 6. Team member contributions

Team Members	Contributions
ROHITH M	Developing UI of the Netbeans, worked on state diagram and design pattern
ROSHAN SRIRAM	Implemented login and register, Transaction class, worked on class diagram and design pattern.
S KALYAN	Implemented booking details class, worked on use case, activity

## 7. Conclusion

So, First we login to the system with username and password that we already data has been fed into the database. Then it takes into another page where we select movie and then after we selecting movie another page comes then we select show timings. After selecting show timings it takes into another page there we select number of tickets we want. After selecting number of tickets it takes into another page there we print ticket with cost.

## 8. References

<https://netbeans.apache.org/kb/docs/java-se.html>

<https://www.mysql.com/>

<https://towardsdatascience.com/10-common-software-architectural-patterns-in-a-nutshell-a0b47a1e9013>