**Software Requirements Specification**

**Movie Ticket Booking System**

Version 1.0

SUBMITTED BY

Monisha S

**Table of content**

1. Abstract

1.1 PURPOSE

1.2 SCOPE

1. Existing system
2. Proposed system
3. Module descriptions
4. Functional requirements
5. Non functional requirements

7.HLD

8.LLD

9.Diagrams: use case,class, sequence,flow chart,entity relationship

10.Test cases

# ABSTRACT

## 

## The project objective is to book cinema tickets in online. The Ticket Reservation System is an Internet based application that can be accessed throughout the net and can be accessed by anyone who has a net connection. This application will reserve the tickets. This online ticket reservation system provides a website for a cinema hall where any user of internet can access it. User is required to login to the system. Tickets can be collected at the counter and watching movies with family and friends in theatres is one of the best medium of entertainment after having a hectic schedule. But all this excitement vanishes after standing in hours in long queues to get tickets booked. The website provides complete information regarding currently running movies on all the screens with details of show timings, available seats. Ticket reservations are done and can be cancelled if needed. Our online tickets reservation system is one of the best opportunities for those who cannot afford enough time to get their tickets reserved standing in long queues. People can book tickets online at any time of day or night. Our reservation system also provides option to cancel the tickets which are reserved previously

## 1.1 PURPOSE:

## The purpose of the online ticket booking system is to provide another way for purchasing cinema tickets in advance. It is an automatic system, we will illustrate our system by providing Use. Case Diagrams with specifications and functionalities of the system. Some process description and data dictionary are provided.

## 1.2 SCOPE:

* The customers can buy ticket online and cancel the seats at a suitable time (2days before the show to 1hour before the show).
* All the customers have to register and become a member before buying tickets.
* Customers canceling tickets will not be given money back instead the amount of money will be recorded in the customer account for further use.
* To make the system more user-friendly, the customer need not enter lots of data.
* Customer will first choose the film time and venue Customer will login to the system.
* Customer will choose the seats positions customers need a credit card to complete a buying transaction.
* Confirm and show the transaction number to the customer.
* In addition to creating a member account, the customer can modify his/her own data.

**3.Existing System**:

Present System developed in asp .In asp there are some limitations.

* It is scripting language
* It is very code complexity
* There is server controls in asp
* It Doesn’t Support language interoperabilty

**4. Proposed System:**

Asp.net Support code behind technology

* It supports different controls like html controls, server controls
* It Supports language interoperability

After understanding the existing system and understanding the need for developing a new system different people involved in the related activities have been consulted. The data needed for the study has been collected from company records. The computerization of this system would avoid the wrong interpretation and bad calculation of data .

The system help the user to see any documents, source code, tasks, activities, team information with details at the click of a button. The record data is maintained and backed up such a way that data is not loss. The speed of the system could also increase.

**4.Modules of Online Movies Ticket Booking:**

**Owner Profile:**

The owner has full access to the system. The owner can graphically view all the details, and he has the authority to change the Cost of tickets, ticket availability, and much more. The owner is provided with an id and password. He can put various kinds of notifications on the website. He can also check the details of seats left and seats reserved.

He can also monitor the transactions made throughout the day and the month. He also can give various kinds of offers to the viewers. He can also keep some seats hidden from the users for their offline reservations. He can add new movies and their show timings. Also, he can delete some videos from the website.

**Viewer Profile:**

In the Viewer profile, the viewer can check the availability of tickets and their category. The category can be silver, gold, or platinum and they can also book their ticket according to their budget and need. Viewer profile contains their name, Contact details, address, and other necessary personal data, etc.

They need to sign-up for booking the movie ticket, which will make them as well as the manager or the owner of the hall interact with each other easily. They can pay the amount Online, and if they need to make payment Offline, they must deposit some advance amount to confirm their movie ticket.

**Manager Profile:**

In the Manager Profile, whosoever is the Manager must log in and then he can book the tickets of the film for the viewers, and this will serve the viewers in offline mode. When he will log-in in to the system, his attendance will be taken, and it will be easy for the owner to monitor them. Here they will also get a notification of booked room so that they can beforehand maintain the place.

**Ticket Booking:**

The viewer can quickly search for the desired movie and the number of seats from the various options available. This all will be so user-friendly that the audience will not find any trouble in booking the movie ticket. Once the available movies are searched, and the viewer finds the movie of his choice then this module helps the audience to book the film.

If everything is done properly, he will be asked to pay partial or complete money. And then he will be forwarded to Payment Gateway where he can quickly pay Online.

**Movie Module:**

This module will help the viewer to see the details of the movies available for the show. He can even search for any specific film. The number of seats available will be displayed to the viewer along with their timings and the cost of different types of tickets. Then he can reserve the required tickets. The information on upcoming movies will also be displayed to the viewers.

**Hall Module:**

It deals with the information about the rooms available for the video shows. Some multiplexes are there. In each multiplex four buildings are there, so the reservation of tickets goes according to the availability of seats in the multiplexes.

**Payment Details:**

After filling in the details like the name of the movie, category of cards, the timings of the show and the number of seats to be reserved, the viewers need to pay the money using different options available like cash/net banking/ATM card/credit/debit card. The payment portal would use a payment gateway to remove the fees.

Once payment will clear receipt of payment will be generated automatically and the message will be sent to the viewer on his mobile. Once payment will clear receipt of payment will be created automatically and the message will be directed to the viewer on his mobile.

**Add/update/delete Movie/booking Details:**

The owner can also easily add, update, or delete the movie, its show timings, and the cost of different categories of the tickets.

**5.Functional Requirements**

This section provide requirements overview of the system various functional models implemented by system

1. User should be able to view the list of movies which are numming near to location based on GPS).

2. User should be able to select the seat as per his choice in the hall.

3. User should have different options of payment.

**Registration**

If the customer want to sea movie then he must be registered unregistered customer can't go for

Movie.

**Login**

Customer login to system by entering valid user id and password

**Changes to Cart**

After the login user can change or cancel the booking

**Payment**

For customer there are many types of secure billing will be prepaid as a debit and credit

**Logout**

After the registration and payment user will be logged out

**Technical Issues**

System work on client Server Architecture it will require internet server

**Interface Requirements**

Various interface for the product could be....

• Login Page

• Registration page

• There should be new, cancel, search options are available.

• Customer information, movies list, ticket price, Seat booking info

**6.Non Functional Requirements**

1. The application should be able to handle 1000 ticket booking requests simultaneously.

2. The application's servers should perform bad balancing efficiently.

3. Application should be lightweight and give quick response.

4. The user's data should be kept safe.

5. The application should be platform independent.

It consist of following parameters.

**Reliability**

The system will consistently performs it indented function.

For e.g The important information must be validated

**Efficiency**

Unnecessary data will not be transmitted on the network and database server will be properly

Connected.

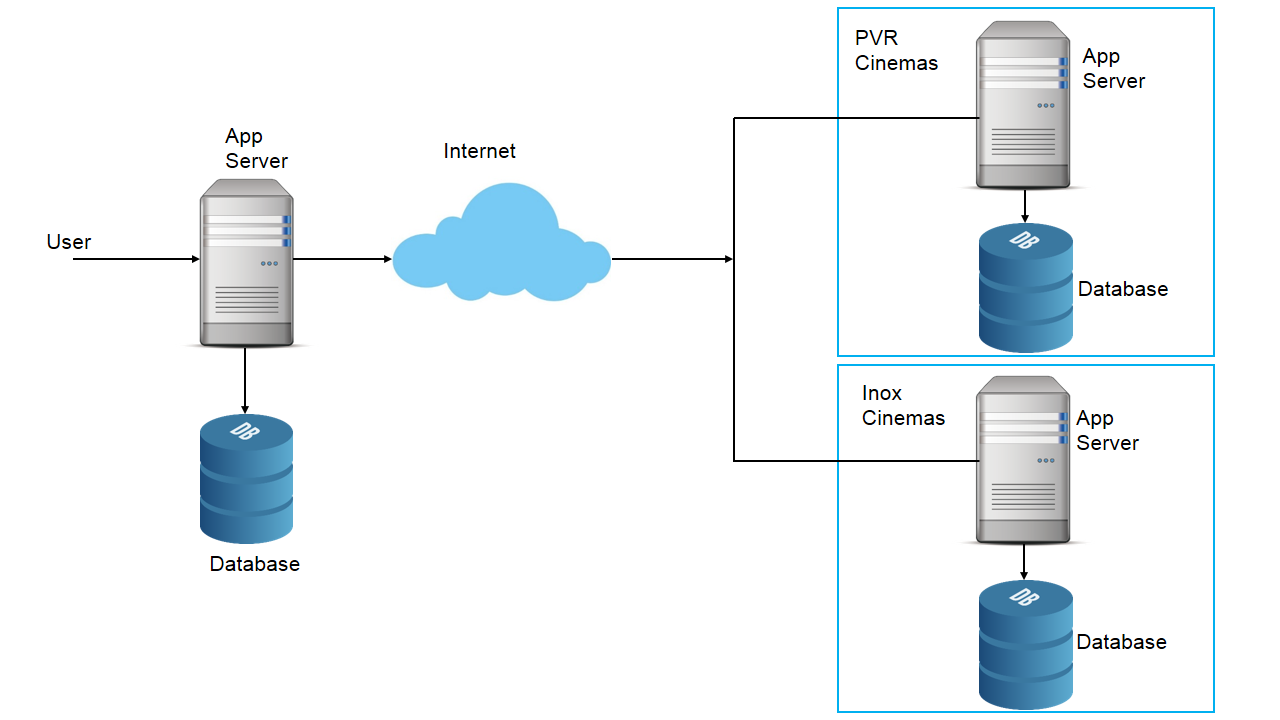
**Reusability**

The system can be used in any Organization or site of the same group.

**Integrity**

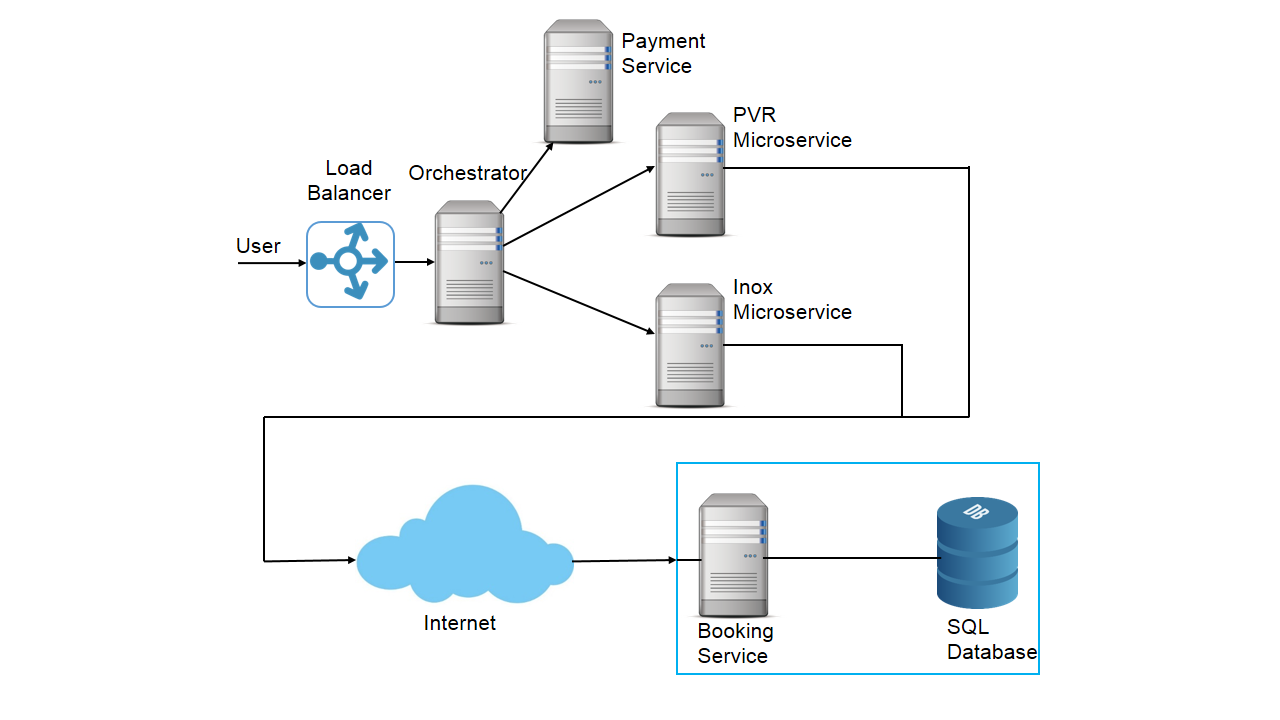
Only system administrator has right to access the database not every user can access all the information.

**7.HIGH LEVEL DESIGN**

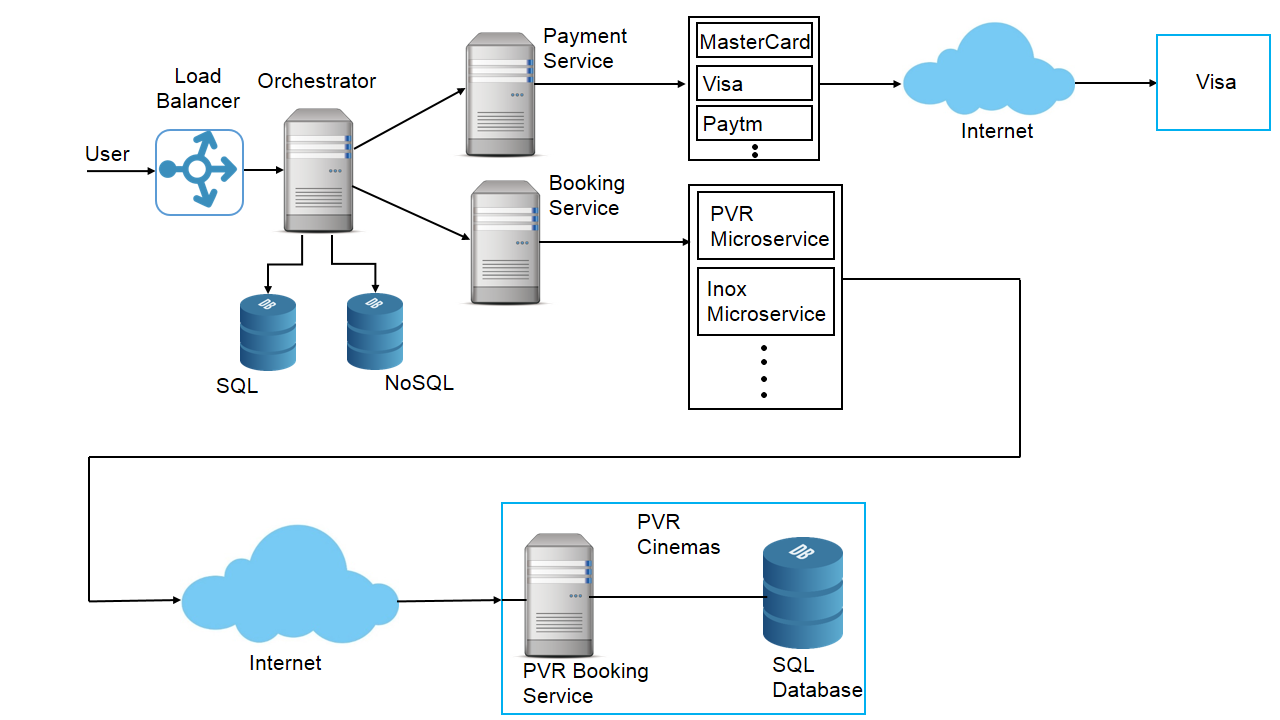


* The user sends request to book a ticket for a PVR movie to our app server.
* Our app server calls the apis of PVR cinemas and requests them to reserve the tickets. This is done so that this seat is locked for the user for say 10 minutes and no other user can reserve it.
* Our app server gets a success response from the PVR cinemas and it redirects the user to the payment page.
* The user makes the payment with the help of the app server.
* Our app server notifies to the PVR cinemas that payment is made.
* The PVR cinemas marks the seat as booked and returns a success response to our app server.
* Our app server notifies the user that the booking is confirmed.
* In case the payment is made and before our app server notifies the PVR cinemas the 10 minute window expires and someone else reserves the seat then we will inform this to our user and refund the amount.

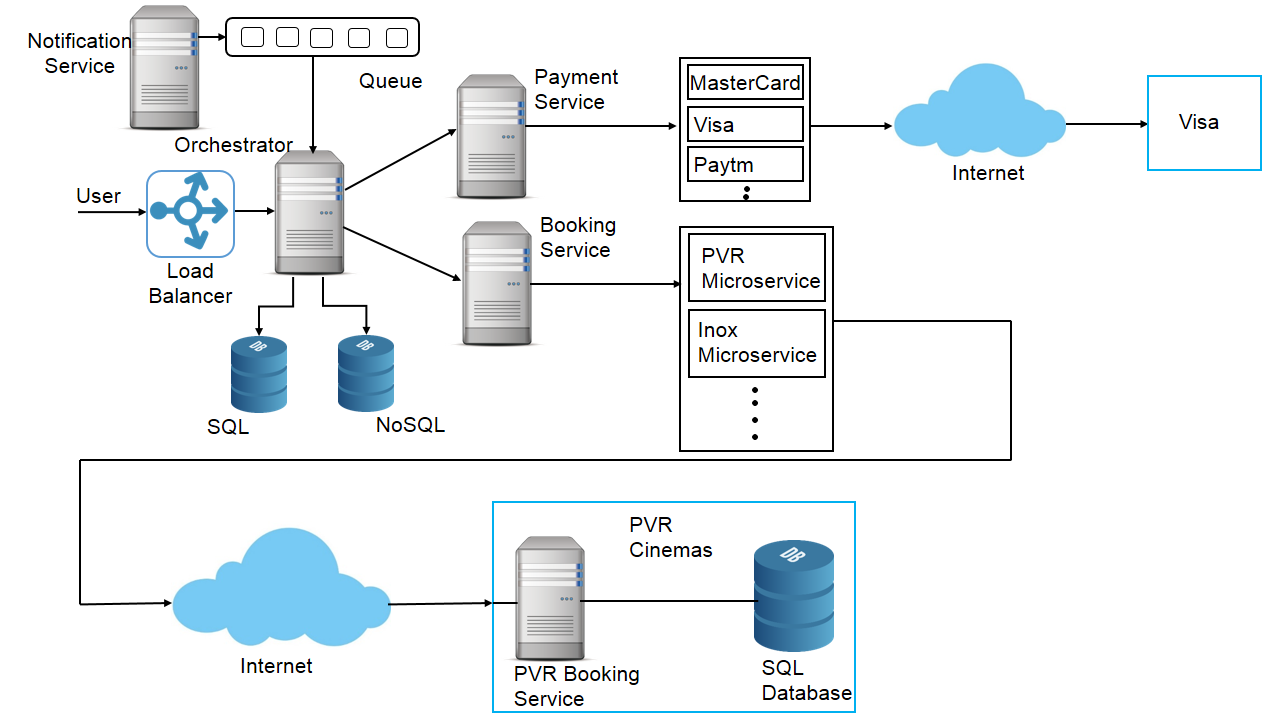
**8.LOW LEVEL DESIGN**

****

**Adding microservice for booking**

****

**Adding microservice for payment**

****

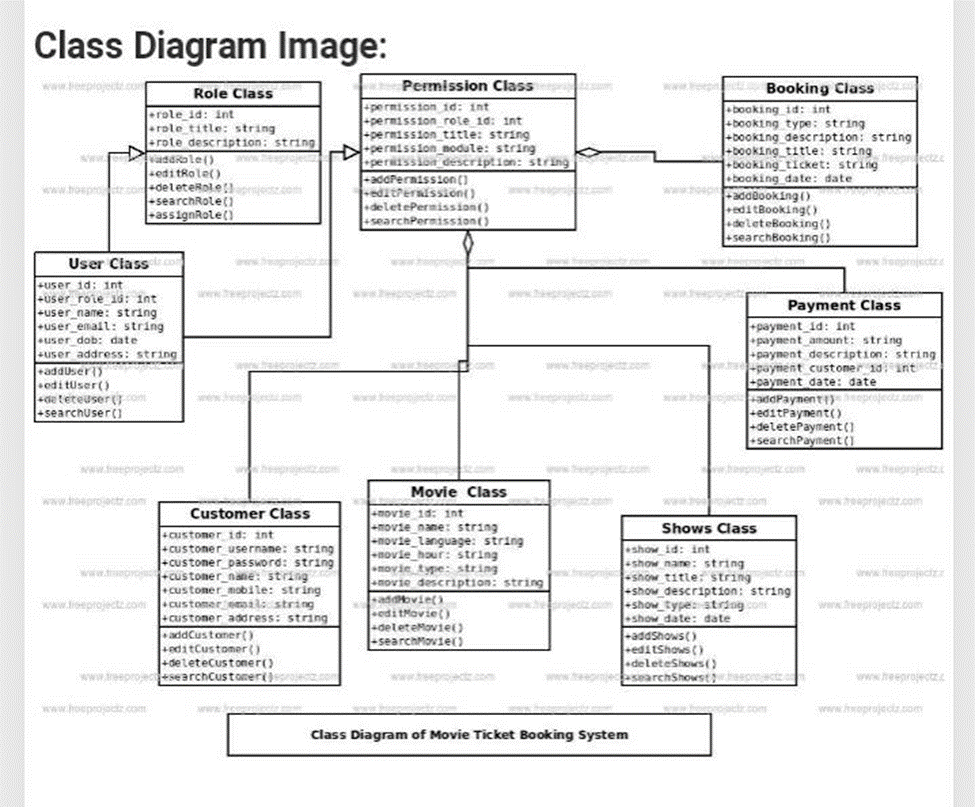
**Adding notification services**

**9.DIAGRAMS**

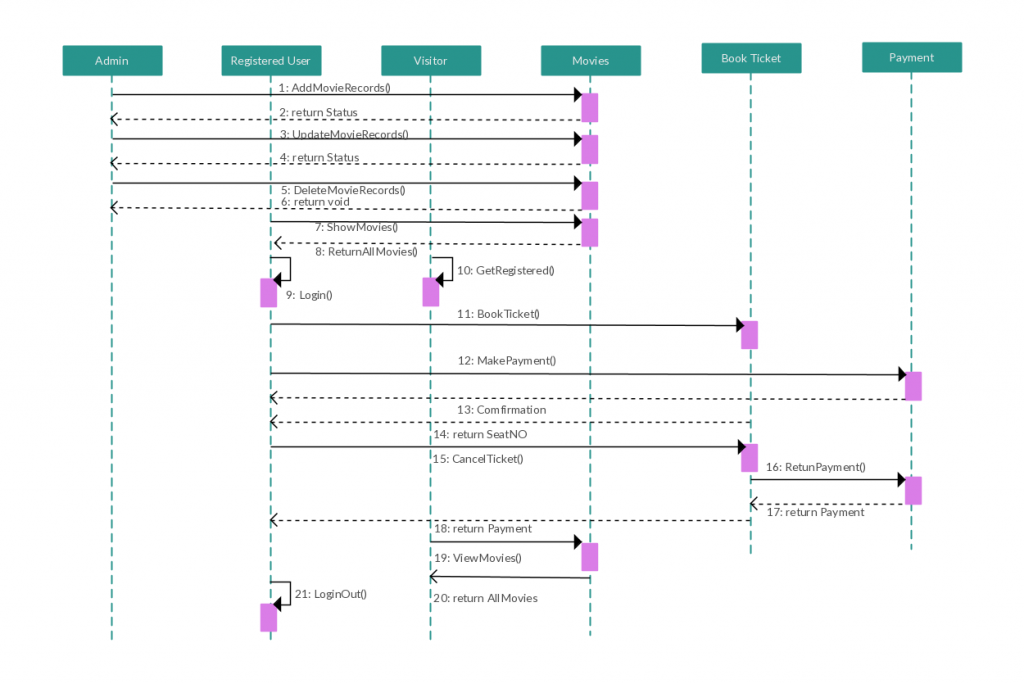
**USE CASE DIAGRAM**



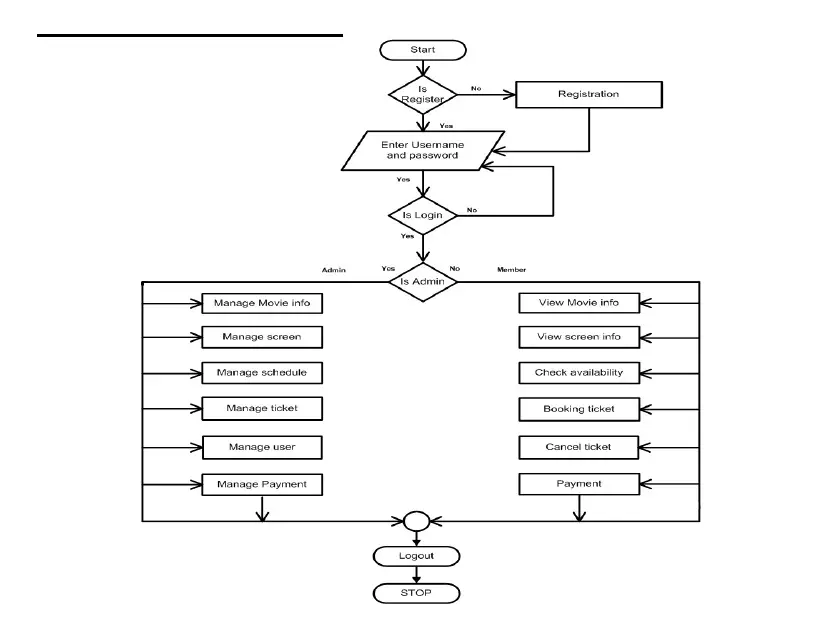
**UML -CLASS DIAGRAM**

****

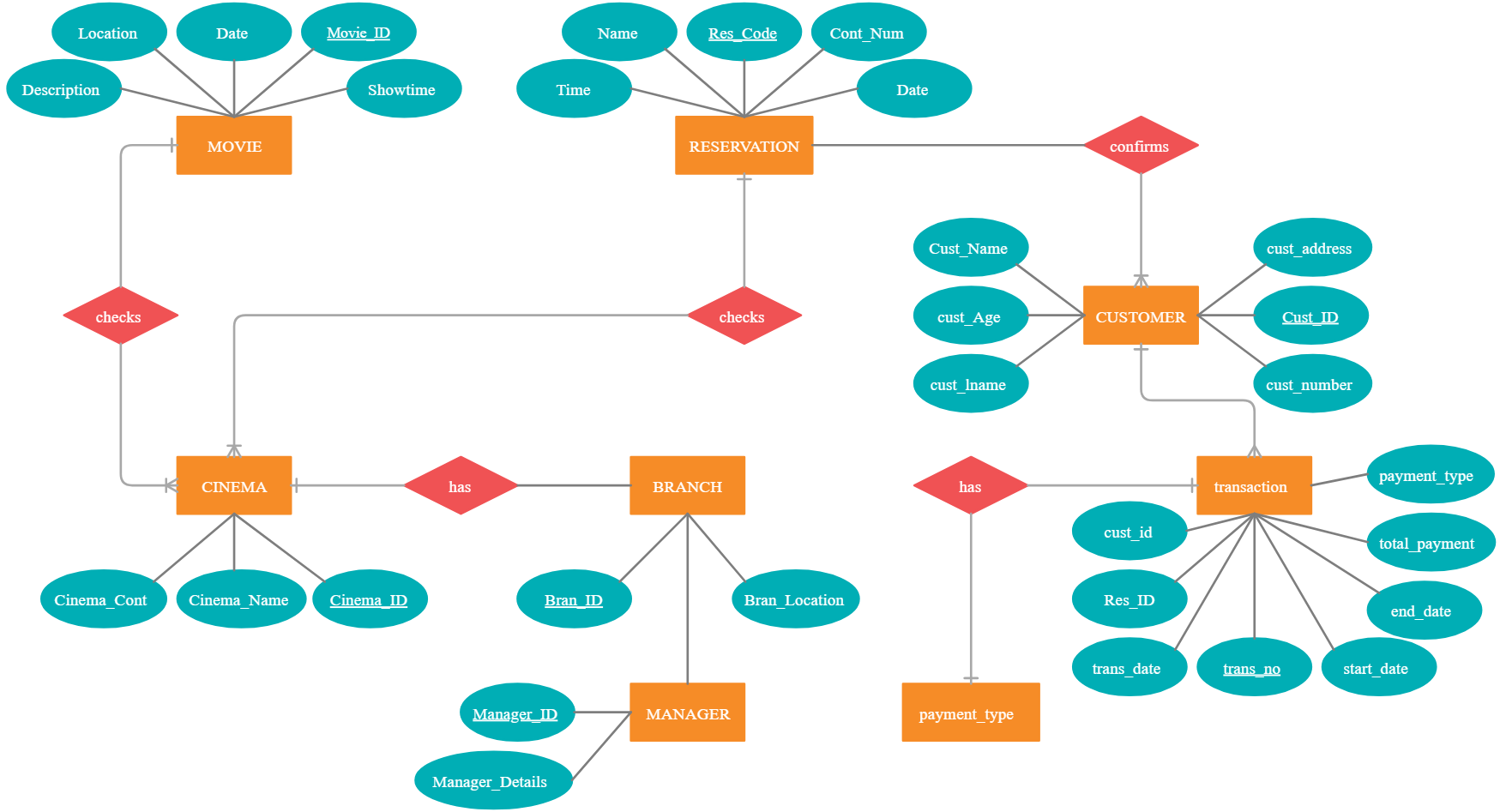
**UML-SEQUENCE DIAGRAM**

****

**FLOW CHART**

****

**ENTITY RELATIONSHIP DIAGRAM**

****

**10.TEST CASES**

The objective is to determine if the software satisfies the business requirements. Besides that, it will identify invalid states and values in the system. After integration of various modules, system integrity will be stressed-tested of its stability. The intended audiences are clients and our supervisor

**Testing strategy**

* both verification and validation test
* constructive test
* testing accomplished by independent test group
* testing conducted independent of debugging
* test scope: unit testing and block testing (due to the limitation of time)

**Software product overview**

The aim of the project is to implement a new online movie ticketing system that enable:

* client to book movie online
* client to cancel movie booked
* client to view information of movie available
* client to choose their own seating
* administrator and user to access the same database through online interface
* administrator to organize movie information online

The advantages of this new system are:

* database system is integrated as part of the online system
* online database does not need to be duplicated for offline database
* clients can pay and book movie through electronics transaction
* clients have flexibility in choosing the seating they like
* administrator can update online (system) database at real time

**Test scope and Test plan**

For unit test, we will employ both static and dynamic testing techniques. Our basic component is the .java and .jsp file. Since the inspection and compilation has already been done by programmer, we will only do the walkthrough for static testing. After that, we will do the white-box testing. The test cases will be derived from basis paths. So they will not be depicted here. Our test completion criteria is the basic function of the module and satisfaction for the requirement.

Forthe high level testing, we divide our program into two major block-customer interface and administrator interface. Instead of the focus on technique aspects, our main concern is the micro-behavior of each block. The black box testing will be employed here. We will use equivalence partitioning, boundary value and logic-based testing techniques for Test Data Selection.