Software Requirement Specification(SRS) for SeatSnap.

1. Introduction

1.1 Purpose:

This document is meant to delineate the features of **SeatSnap**, so as to serve as a guide to the developers on one hand and software validation document for the prospective client on the other.

A Website offers user to book ticket over internet. The **SeatSnap** provides complete functionality of listing and booking movies. It also offers comprehensive information on currently running movies including details on show timings, available seats and seat costs.

1.2 Scope:

This system encompasses various aspects related to providing personalized movie recommendations to users.

1.3 Definitions:

SRS- Software Requirement Specification

GUI- Graphical User Interface

DFD - Data Flow Diagram

UML - Unified modelling Language

1.5 Overview:

SeatSnap is an online movie ticket booking system that allows users to book tickets over the internet. To access the system, users need an internet connection and must log in.

Upon Successful booking, a copy of the tickets is sent to the user via email and SMS. The website offers comprehensive information on currently running movies across all screens, including details on show timings, available seats, and seat costs.

2.Overall Description:

The SeatSnap application enables admin to add a movie, theatres and manage showtime and allows user to book movie tickets. Developer is designing an online movie-ticket booking system to manage the movies in the portal and also help customers to book them online without visiting the movie center physically. The SeatSnap will use the internet as the sole method for booking movies for customer.

2.1 Product Perspective:

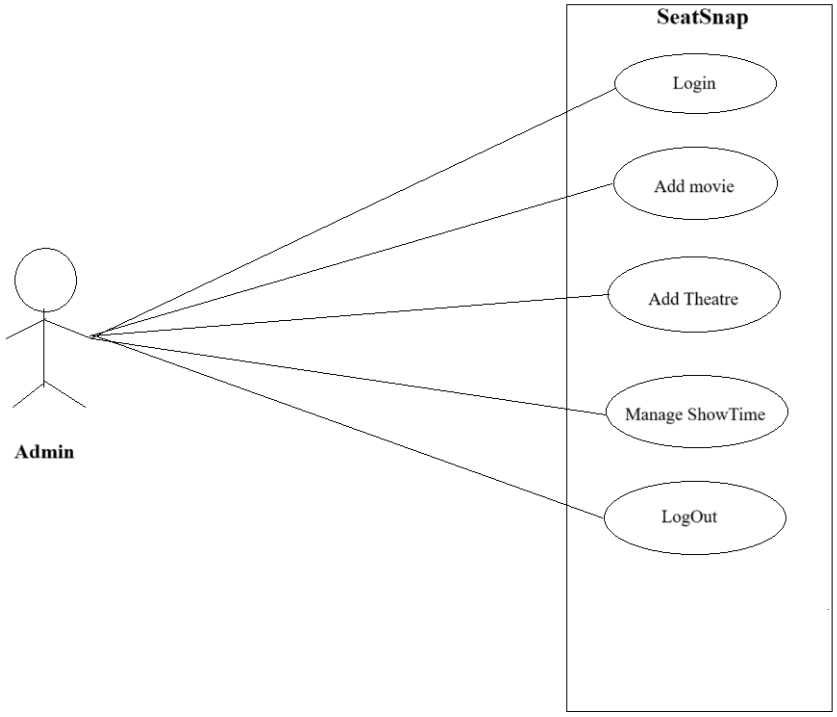
This product aimed toward a person who don't want to visit the movie center as he might don't get time for that or might not interested in visiting there and dealing with lot of formalities.

2.2 Product Functions:

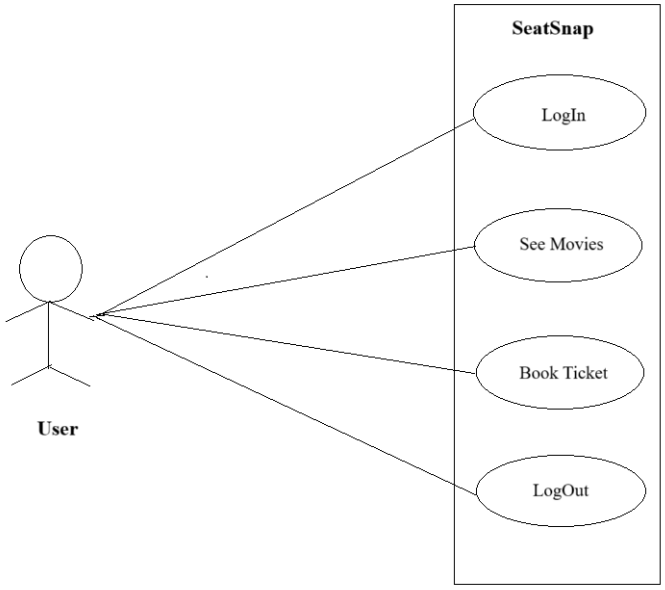
SeatSnap should support this use case:

Use Case Diagrams : A Use case is a description of set of sequence of actions. Graphically it is rendered as an ellipse with solid line including only its name. Use case diagram is a behavioral diagram that shows a set of use cases and actors and their relationship. It is an association between the use cases and actors. An actor represents a real-world object. Primary Actor - Sender, Secondary Actor - Receiver.

Use case diagram for admin



Use Case diagram for User



2.3 User Characeristics:

User should be familiar with the terms like login, register and booking etc.

2.4 Principle Actors:

2 Principle Actors are Customer and Administrator.

2.5 General Constraints:

A full internet connection is required for SeatSnap.

2.6 Assumptions and Dependencies:

Working of SeatSnap need Internet Connection.

3. Specific Requirements:

3.1 FUNCTIONAL SPECIFICATION

User Specification

Admin:

Admin can add a movie, theatre and manage show time and also view the booking.

User:

User can view current streaming movies and book tickets.

MODULE SPECIFICATION

User

•View Current movies:

The user can view current movies and user can book seat.

•Booking Ticket:

The user can view current streaming movies and user can book.

•Easily Get the Seat:

The Customer can easily get the seat if available whenever they need with use of this system.

Admin

Dashboard:

In this section admin can view the overview of the SeatSnap (Like theatre, movies, seats, time etc.)

Movies:

The Admin can add the movie so that the user can see the available movies and book the seat for it.

Admin can also edit and delete the showtime of movie.

Manage Contact us query:

Admin can manage **Contact Us** query.

3.2 Non-Functional Requirements:

Following Non-Functional Requirements will be there in the insurance to the internet:

(i) Secure access to consumer's confidential data.

(ii) 24X7 availability.

(iii) Better component design to get better performance at peak time.

(iv) Flexible service based architecture will be highly desirable for future extension. Non-Functional Requirements define system properties and constraints.

Various other Non-Functional Requirements are:

- Security
- Reliability
- Maintainability
- Portability
- Extensibility
- Reusability
- Compatibility
- Resource Utilization

3.3 Performance Requirements:

In order to maintain an acceptable speed at maximum number of uploads allowed from a particular customer as any number of users can access to the system at any time. Also the connections to the servers will be based on the attributes of the user like his location and server will be working 24X7 times.

3.4 Technical Issues:

This system will work on client-server architecture. It will require an internet server and which will be able to run web application. The system should support some commonly used browser such as Edge, Mozzila Firefox,Google Chrome etc.

HARDWARE REQUIREMENT

Hardware requirements for insurance on internet

will be same for both parties which are as follows:

RAM	2 GB
Hard disk	320 GB
Processor	Dual Core

Software Requirements

Client side:

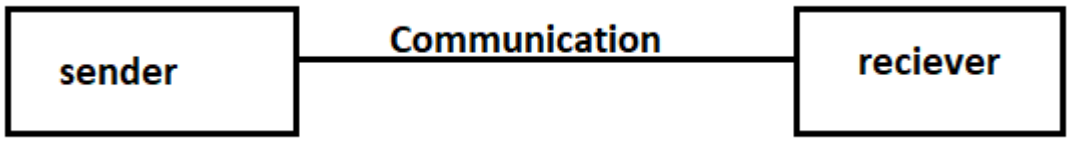
Web Browser	Google Chrome or any compatible browser
Operating System	Windows or any equivalent OS

Server side:

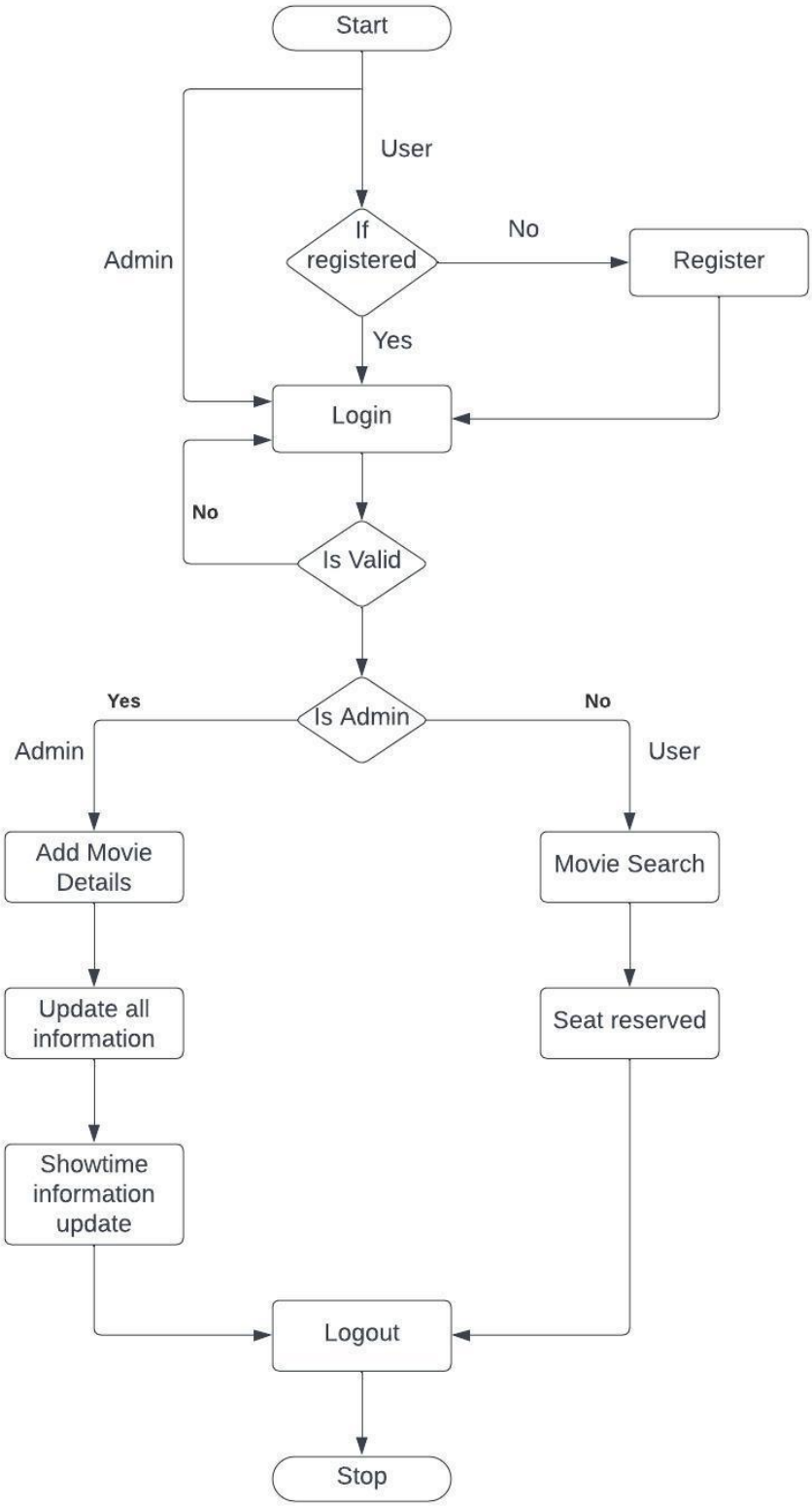
Web Server	TOMCAT
Server side Language	JAVA
Database Server	MYSQL
Web Browser	Google Chrome or any compatible browser
Operating System	Windows or any equivalent OS

Communication Interfaces:

The two parties should be connected by LAN or WAN for the communication purpose.



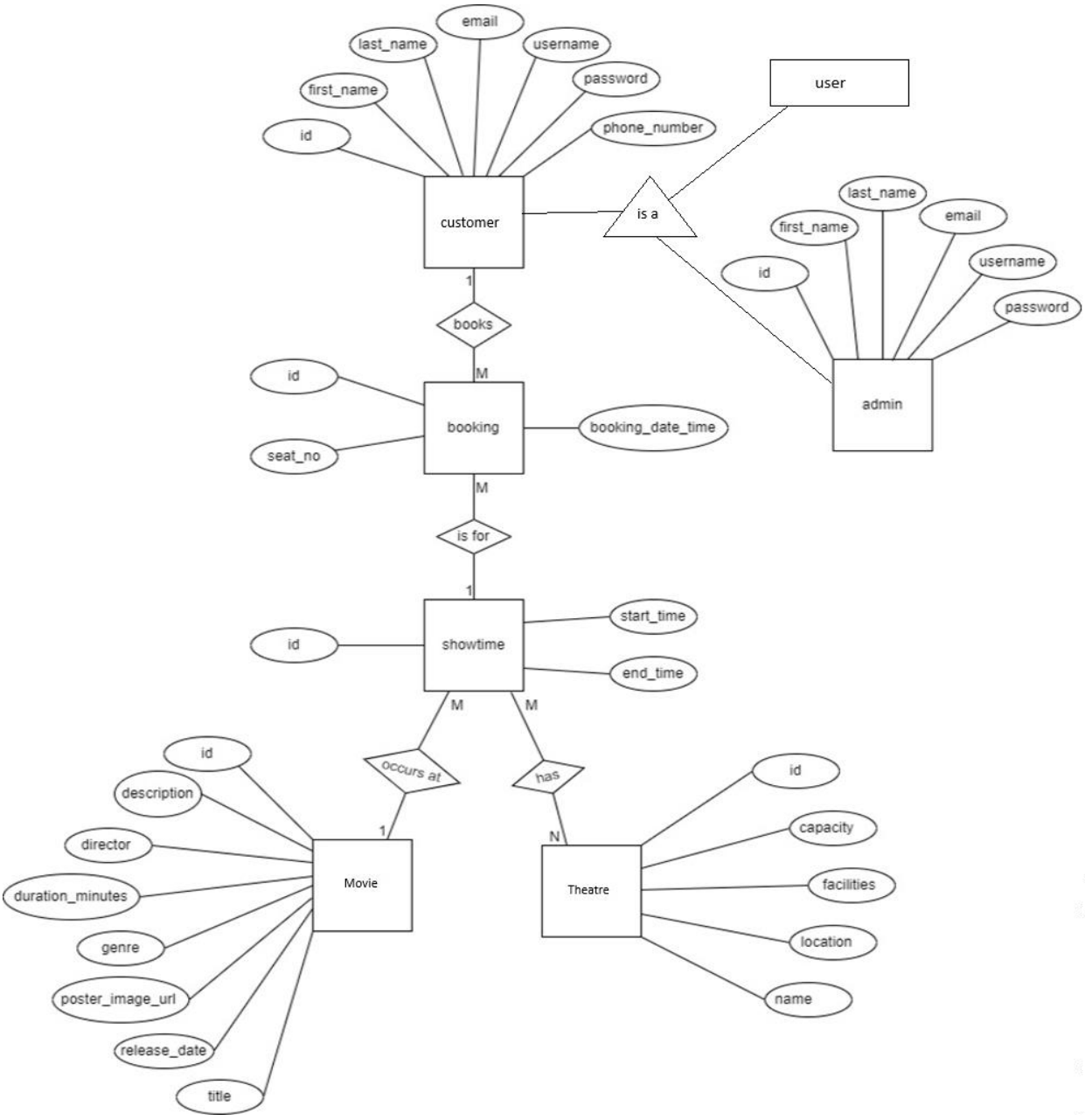
5.System Design Specification: System Flow Chart



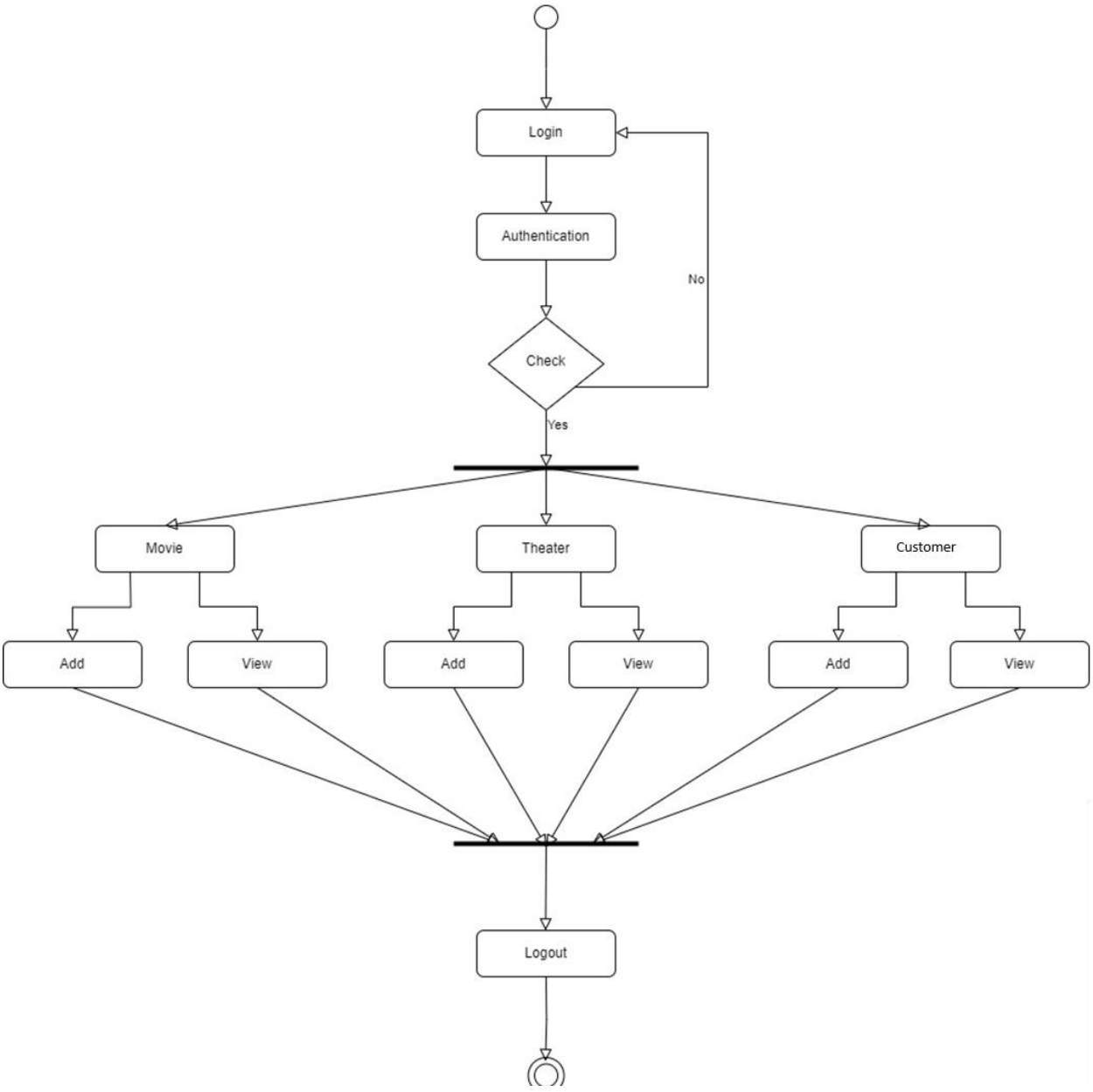
ER DIAGRAM

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design for the database designer, the utility of the ER model is:

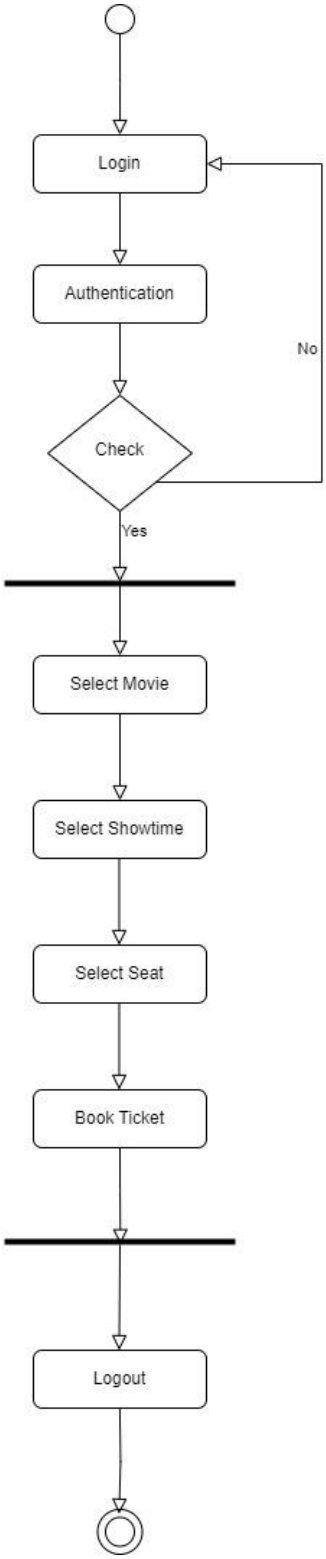
- It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
- It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user.
- In addition, the model can be used as a design plan by the database developer to implement a data model in specific database management software.


email
last_name
username
password
first_name
phone_number
id
user
customer
is a
last_name
email
first_name
username
id
password
admin
1
books
M
id
booking
booking_date_time
seat_no
M
is for
1
id
showtime
start_time
end_time
M
M
occurs at
has
id
description
director
duration_minutes
genre
poster_image_url
release_date
title
1
Movie
N
Theatre
id
capacity
facilities
location
name

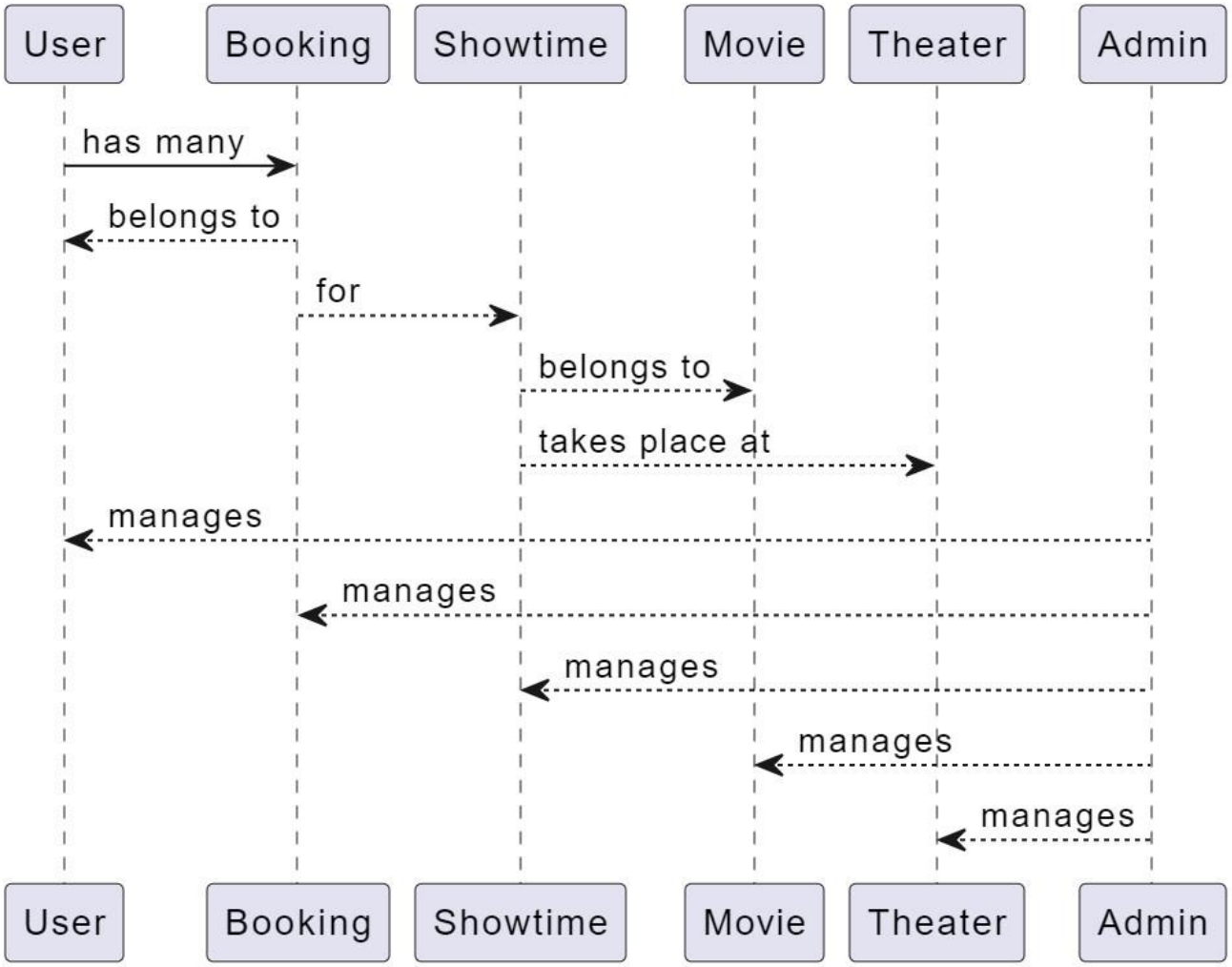
Activity Diagram for admin



Activity Diagram for user



Sequence Diagram



DATABASE DESIGN

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and

optimizing for updates. The MySql database has been chosen for developing the relevant databases.

Tables:

SeatSnap
admin
booking
movie
showtime
theater
user

Admin:

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
email	varchar(255)	YES	UNI	NULL	
first_name	varchar(255)	YES		NULL	
last_name	varchar(255)	YES		NULL	
password	varchar(255)	YES		NULL	
username	varchar(50)	YES	UNI	NULL	

Booking:

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
booking_date_time	datetime(6)	NO		NULL	
seat_no	varchar(300)	NO		NULL	
showtime_id	bigint	YES	MUL	NULL	
user_id	bigint	YES	MUL	NULL	

Movie:

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
description	varchar(1024)	YES		NULL	
director	varchar(255)	YES		NULL	
duration_minutes	int	NO		NULL	
genre	varchar(255)	YES		NULL	
poster_image_url	varchar(255)	YES		NULL	
release_date	date	NO		NULL	
title	varchar(255)	YES		NULL	

Showtime:

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
end_time	datetime(6)	NO		NULL	
start_time	datetime(6)	NO		NULL	
movie_id	bigint	YES	MUL	NULL	
theater_id	bigint	YES	MUL	NULL	

Theater:

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
capacity	int	NO		NULL	
facilities	varchar(1024)	YES		NULL	
location	varchar(255)	YES		NULL	
name	varchar(255)	YES		NULL	

User:

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
email	varchar(255)	YES	UNI	NULL	
first_name	varchar(255)	YES		NULL	
last_name	varchar(255)	YES		NULL	
password	varchar(255)	YES		NULL	
phone_number	varchar(255)	YES	UNI	NULL	
username	varchar(50)	YES	UNI	NULL	