Movie Reservation Website

General Description

The proposed web application allows the user to book tickets to a movie theater. The user can:

1. Select the movie from the predefined list.
2. Choose the desired date in the calendar.
3. Choose the place in the cinema hall.
4. Type own name and confirm own choice with OK button.
   (The system should notify whether the reservation is completed successfully.)
5. Additionally, the user can register in the system, so that the app will not have to ask the username during the next booking.
6. (Optional) The system can show the history of movies booked by the currently logged user.

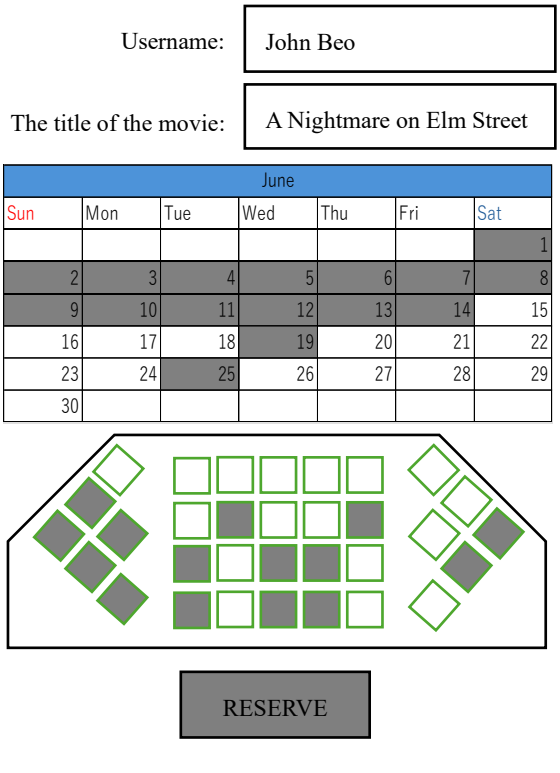
Notes:

1. There are 10 different movies in the system.
2. There are 5 movies shown daily in the theater.
3. Reservation can be done for the next 14 days.
4. The number of seats in the hall is limited, and they are different.
5. All unavailable days in the calendar, and all unavailable seats in the cinema hall should be grayed out (i.e., disabled). If no movie is chosen from the list, or no date is chosen from the calendar, all the days should be grayed out.
6. The system should identify the situation when two users are trying to reserve the same seat simultaneously and prevent overbooking. (The system should print the message “Please choose other seats, or other day”.)

Requirements for rich user interface:

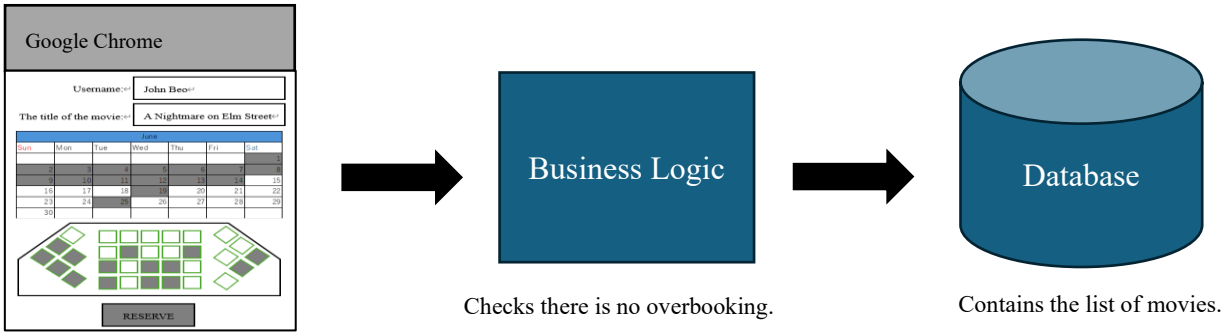
1. If the user selects the desired date first, he/she can only select movies available for this date.
2. If the user selects the desired movie first, he/she can only select dates when this movie is shown.
3. If there are no available seats for a certain day/film, the app should inform the user about it.
4. The user should be able to select the seat using the movie hall map. (Available seats should be highlighted.)
5. Registration requires entering a unique combination of login/password.
6. If a certain user is logged in, there is no need to ask for a username.

User Interface (Sketch)



Proposed Architecture

The system will be implemented using client-server architecture. It will include the elements shown in the picture below. We will use Django framework for building the main app logic, and Vue.js for the rich user interface.



Frontend (user interface)

Checks there is no overbooking.

Stores user reservation in the database.

Contains the list of movies.

Keeps Reservation.

Database Structure

The database will contain three tables, movies, showtimes, seats, and users, having the following structure.

Table movies

Column	Type	Comment
id	int	Unique movie ID
name	char(20)	Movies name

Table showtimes

Column	Type	Comment
id	int	Unique showtime ID
time	date	Date of this schedule
number	int	The n-th movie for this date (n is from 1 to 5)
movie	int	ID of the movie (foreign key)

Table seats

Column	Type	Comment
id	int	Unique seat ID (shows the number of the place of the seat)
showtime	int	ID of the showtime (foreign key)
reserved	int	This can take 0 (not reserved), or 1 (reserved)
user	int	ID of the user (foreign key)

Users

Column	Type	Comment
id	int	Unique user ID
name	char(20)	Username

Project Plan

1. Setup the system (all required software tools).
2. Create the database and design its structure.
3. Create the business logic. Make sure it works without user interface.
4. Create the prototype of the user interface in plain HTML.
5. Create a rich interface in Vue.js.