

Project Description

Boris Glavic

Project Timeline

The project has three deliverables. Deadlines are announced on the course webpage <https://www.cs.uic.edu/~bglavic/cs480/2026-spring/schedule/importantdates/>. This document can be downloaded from <https://www.cs.uic.edu/~bglavic/cs480/2026-spring/pdfs/project.pdf>. Each group will **demo** their application at the end of the semester. The deliverables are:

- **ER-model:** Each group should develop an ER-model for the application. This can be uploaded as any type of image file (please do not use esoteric formats).
- **Relational schema:** The second deliverable is a translation of the ER-model into a relational schema implemented as an SQL script. The script should use Postgres's SQL dialect. Besides from defining tables and constraints, this script should create indexes where appropriate. Please upload the script as a simple text file. You can also add example test data.
- **Application:** The last deliverable is an online grocery management application that uses the relational schema defined in the first two deliverables. This application can be either a web or desktop application.

Some of the requirements are marked as optional **bonus** requirements. You are free to not realize these requirements, but you can get extra credits by implementing them. **Every member of the group has to contribute in each phase of the project** and you will be graded based on your individual contribution and on the overall project result.

Overview - Movie Ticketing System

In this project you are building a system for a cinema which enables the cinema to keep track of movie scheduling, ticket sales, and their clients.

Data Requirements

Users

The system should support two types of users.

Clients

Clients are the clients of the cinema. We want to record their name, address, email address, and (optionally) a short description of their movie interests. Email addresses are unique. Some clients are signed up for a reward program, that gives them a free ticket for every 10 movies they watch.

1. Payment methods

For clients we record payment methods. A payment method is either a credit card or a debit card where we have to record the credit card number and billing address as well as for a debit card just the number.

Administrators

For administrators we only record necessary information for their login (an email address which is unique).

Theaters

For a **theater** we want to record an id and a maximum occupancy (available seats). Furthermore, only some theatres are *3D theatres* and some theaters have a *fancy sound system*.

Schedule

The **schedule** for the cinema records for which date and time which movie is shown in which theater. The cinema has regular opening hours and movies can only be shown during the opening hours. Obviously, at each time only a single movie can be shown in a particular theater.

Movie ticketing

For each **movie screening** (timeslot at a particular theater where the movie is screening), we record ticket sales. For each screening we cannot sell more tickets than the maximum occupancy of the theater in which the screening takes place. The price of a ticket is determined as follows:

- There is a base price for tickets of \$15
- If the screening is in a 3D theater then the price is increased by \$5
- If the screening is in a theater with a fancy sound system then the price is increased by \$3
- If the movie is by a major studio then the price is increased by \$3
- If the movie is old then the price is decreased:
 - if the release date more than 2 months in the past, then the final price is reduced by 20%
 - if the release date more than 2 years in the past, then the final price is reduced by 40%

A single **ticket sale** can be for multiple tickets for the same screening. Tickets can either be brought by registered clients (in which case we need to record which registered client brought the ticket for reward purposes) or by an anonymous client (in this case we have no information about who brought the ticket). When a ticket is bought by a registered client then we need to record which of the client's payment method was used.

Movies

For **movies** we would like to record several pieces of information:

- **Title**
- **Major studio:** was the movie produced by a major studio or not
- **Release date**
- **Length**
- **Original language**
- **Available languages**
 - one or more languages
- **Directors**
- **Actors**

- We need to store which actor played which character (name) in the movie. An actor may play multiple characters in one movie.
- **Writers**
- **Producers**

Persons

A single **person** can take on different roles in different movies (actor, director, ...). We would like to store information about a person. Such a person can then participate in different roles in different movies. For a person we would record their

- **name**
- **birthdate**
- **biography** (text)
- **awards**
 - A person can have any number of awards (title, year, and for what role in which movie, e.g., "Oscar" 1992 for being the "director" of movie "X").

Application Requirements

User registration & login

The system should allow users (clients and admins) to register using their email address, setting a password for login. Registered users can log into the system using their password. Based on the type of user, they have different functionality available to them.

Clients

1. Ticket booking

Clients should be able to search for available tickets for movie screenings, searching by the movie title and other movie metadata, as well as timeslots. For example, the client may want to book a screening for a drama movie on Saturday 02/21 after 8 p.m. The system should show the client movies matching their search (showing the movie information such as title, actors, ...). For each movie we should show the theaters that should the movie for a timeslot matching the clients search criteria, how many seats are available in each theater, the price per seat, and whether the theater is 3D or features a fancy sound system. The client can then select a screen and book number of tickets (of course not more than the currently available number of seats). **The system should support concurrent ticket booking by multiple clients.**

2. Managing payment methods

Clients should be able to manage (add, delete, update) their payment methods.

Administrators

1. Movie screening scheduling and monitoring

Administrators can schedule movie screenings. Furthermore, the system should enable them to see the current remaining available seat capacity of each movie screening.

2. Ticket sales & occupancy analytics

To support future planning, the system should provide the administrator with reports showing important ticket sales and occupancy statistics. The precise set of statistics to use is up to

you, but here are some relevant ones:

- The revenue (sum of prices of sold tickets) broken down by time and/or theater and/or movie
- The average, minimum, maximum occupancy rate (percentage of occupied seats) for a given time interval and/or theater

BONUS: RAG-based movie recommendation system

The cinema wants to recommend movies to its clients using **RAG** (retrieval-augmented generation). The plan is to embed movie information (title, descriptions, ...) using an embedding model. Clients registered with the cinema can submit a description of their interests to the cinema which will also be embedded. The embeddings should be stored in Postgres using the `pgvector` [extension](#) which allows you to store vectors (resulting from employing an embedding model) as values in postgres and to build indices (HNSW) for approximate nearest neighbor (ANN) search. To produce a recommendation for a client, you need to embed the clients description of interests and then find the 3 most similar movies in the embedding space as a recommendation.

BONUS: Movie scheduling with occupancy prediction

The movie cinema has to schedule movies to theaters. An optimal schedule would assign movies to theaters such that maximum occupancy is achieved (ideally all theaters are as full as possible) without running out of capacity for theater rooms (not all clients who want to see a movie are able to get a seat). Of course, optimal planning would require knowing the future which is not possible. Instead the theater wants to use a machine learning regression model to predict future seat requirements for movies and then model optimal scheduling as an optimization problem.

Seat requirement prediction model

For the prediction model you have to decide what features to use for prediction. These could include time-based features (more people will go to the movies on a Saturday evening than Wednesday morning), movie metadata (genre, director, budget, ...), and past occupancy data. You can decide what model you want to train, but given that this is not the main purpose of the project, it may be ok to go with a simple model, e.g., random forest. Given the features, the model should predict the number of tickets that could be sold for this movie and timeslot.

Optimal scheduling

For scheduling, the theater assumes that the prediction model is correct. The goal is to plan for the next week in the future to maximize expected revenue (tickets sold) by assigning movies to theaters and time slots. For predicting revenue for a candidate schedule, you should assume that if a movie m is scheduled at time slot t , then the number of tickets sold should be equal to the occupancy predicted by the model for this movie and timeslot capped at the seat capacity of the cinema. Recall that the price of a ticket is determined based on the theater category and movie type (new big studio movie, new movie, old movie).

A solution to this optimization problem has to obey the following constraints:

- During each time slot only one movie can be shown in each theater