VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY

Ho Chi Minh City University of Technology

Faculty of Computer Science and Engineering

DATABASE SYSTEMS (CO2013)

Assignment 1

BKinema
A Movie Ticket Booking System

Instructor: Le Duc Hoang Nam

Semester: 251
Class: CC06
Group: BKinema
Students: Nguyen Trung Nhan – 2352852
Le Pham Tien Long – 2352688
Ho Minh Nhat – 2352858
Nguyen Vinh Phu – 2352919
Pham Nam An – 2353016

HO CHI MINH CITY, OCTOBER 2025

Contents

1 Introduction

In recent years, the global film industry has witnessed rapid digital transformation, reshaping the way audiences engage with cinema. With the increasing penetration of the Internet and the proliferation of smartphones, online movie ticket booking platforms have become an essential part of modern entertainment consumption. Audiences now expect seamless, real-time access to film schedules, personalized recommendations, and convenient booking experiences without the need to queue at traditional box offices. According to a report by SkyQuest Technology Consulting (2025), the global online movie ticketing services market was valued at USD 23.14 billion in 2023 and is projected to grow to USD 39.43 billion by 2032, with a compound annual growth rate of 6.1% during the forecast period 2025 – 2032 [1].
In Vietnam, similar trends have emerged as digital platforms increasingly dominate the entertainment landscape. Leading cinema chains such as CGV Cinema, Lotte Cinema and Galaxy Cinema have developed advanced online booking systems that allow users to browse movies, select showtimes, reserve seats, and complete payments through secure online channels.
Statistical reports show that Vietnam's cinema market is experiencing strong post-pandemic recovery and digital growth. Total box office revenues reached approximately around 185 million USD in 2024, the highest ever recorded in the country [2]. The number of cinema complexes tripled over the last decade, rising from 79 (in 2014) to 212 (in 2023) [3]. With over 79.8 million Internet users (78.8% penetration) and 127 million mobile connections nationwide [4] in the early of 2025, the potential user base for online movie ticketing is remarkably large.
Inspired by these existing systems, the proposed application **BKinema** aims to provide a user-friendly platform for online movie ticket booking.

Figure 1: *BKinema logo.*

By analyzing current digital cinema platforms and integrating advanced data management concepts, this project seeks to design a robust and scalable database system for BKinema. This database will form the backbone of the application, supporting essential operations such as movie catalog management, user authentication, booking transactions, and recommendation algorithms. Ultimately, BKinema aspires to enhance the overall digital cinema experience through intelligent design, efficient data processing, and user-centered interaction, contributing to the ongoing digital transformation of Vietnam's entertainment industry.
The remaining of the report is structured into four main sections. Section 2 presents a data requirements analysis, identifying essential entities, attributes, and relationships within the BKinema system. This section also examines reference systems such as CGV to derive practical insights. Semantic constraints that cannot be represented directly in the Entity–Relationship Diagram (ERD) model are carefully discussed to ensure logical consistency in the design. Section 3 illustrates the proposed ERD, providing a conceptual visualization of the BKinema database. The model incorporates entity hierarchies, weak entities, multivalued and derived attributes, as well as identifying and recursive relationships. Section 4 focuses on the database schema mapping process, translating the conceptual EER model into a relational schema. It explains design decisions such as normalization, integrity constraints, and the creation of tables for relationships and subclasses to ensure data integrity and scalability. Finally, Section 5 summarizes this study's key findings and outlines possible directions for future implementation.

2 Data Requirements Analysis

2.1 Research and Reference System

2.1.1 Booking Ticket and F&B

The CGV Cinema system was selected as the reference model for our project due to its complete and well-structured online booking workflow. The process is designed to guide customers through a sequence

of logical and user-friendly steps, from browsing movies to making payments. The interface is responsive and consistent across web and mobile platforms.

- **Browse Movies:** Users can explore currently trending and upcoming films with detailed descriptions, trailers, and genres. The homepage (Figure 2) provides featured movies, promotions, and quick navigation options.

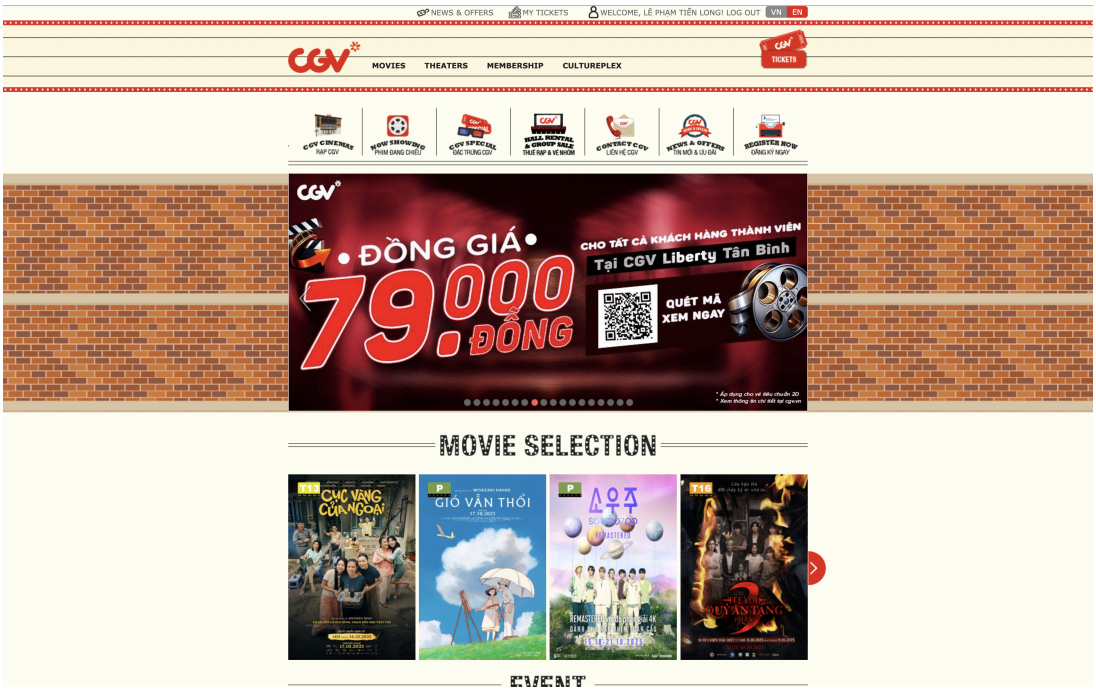

Figure 2: *Homepage displaying featured movies and ongoing promotions.*

- **View Showing Films:** This screen (Figure 3) lists all movies currently available for booking. The list is dynamically updated based on cinema branches and screening dates, ensuring users only see valid and available showings.

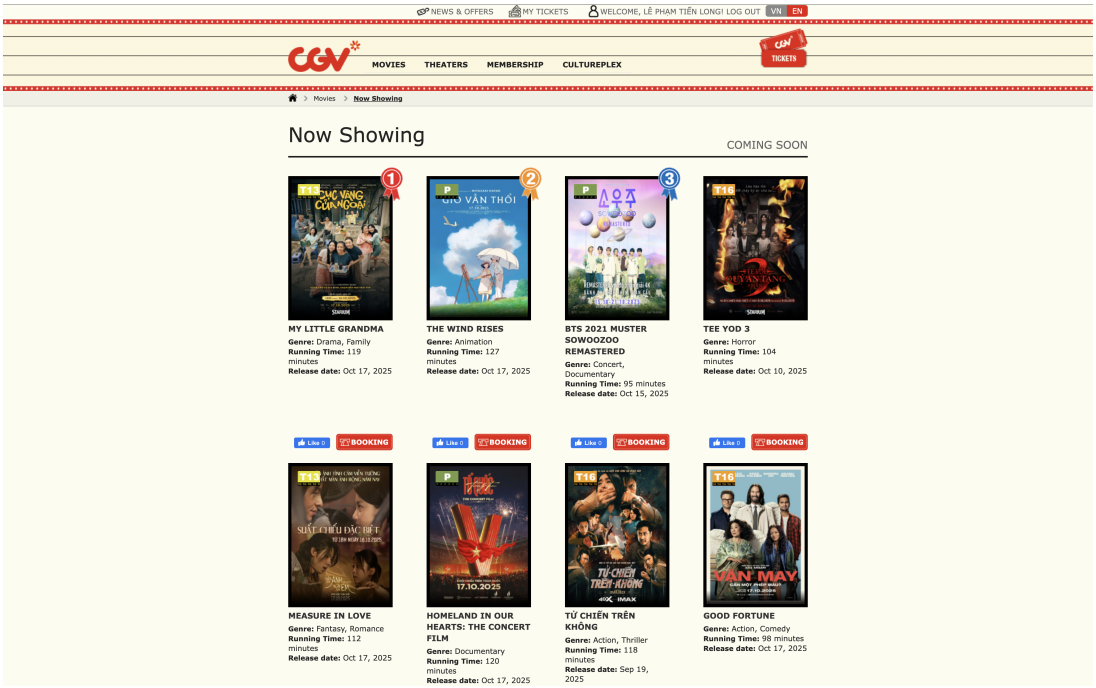

Figure 3: *List of movies currently showing in the CGV system.*

- **Choose Movie:** By selecting a movie, users can access its synopsis, cast, runtime, and ratings (Figure 4). The system cross-checks the user's age eligibility with the movie's *age_rating* constraint before allowing a booking. Additionally, the movie page dynamically retrieves data such as real-time seat availability and showtime schedules.

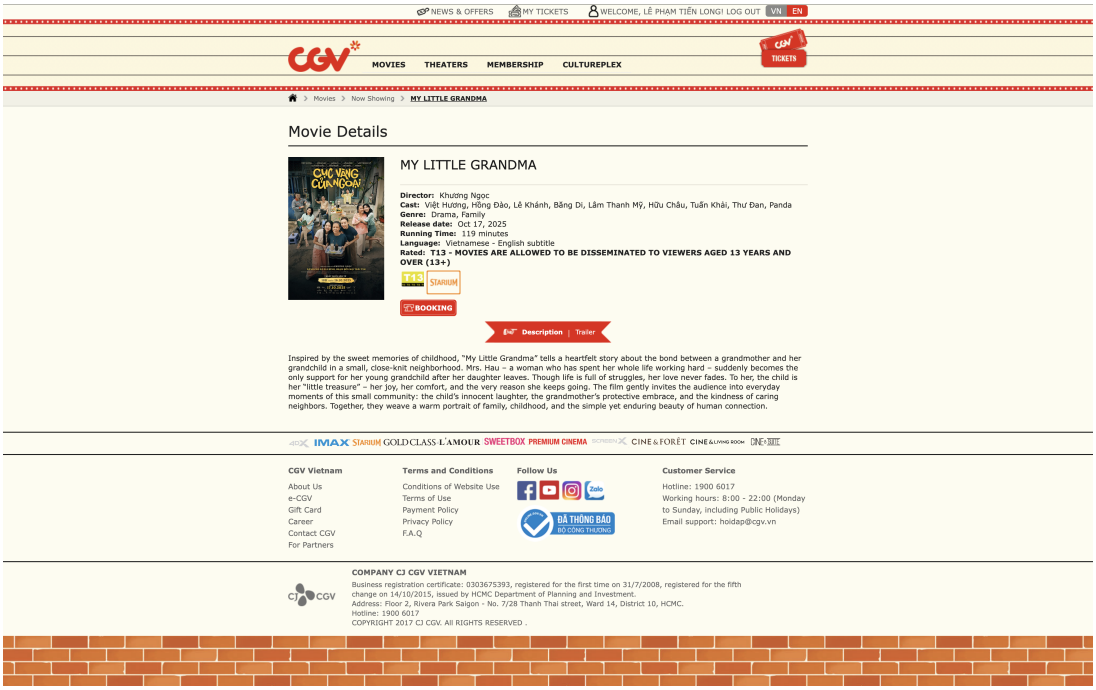

Figure 4: *Detailed movie information and schedule overview.*

- **Select Showtime:** Users can specify the cinema branch, preferred date, and available showtime. As shown in Figure 5, the system ensures that no booking can be made for showtimes that have already ended, and that the time gap between two sessions in the same hall is at least 30 minutes for cleaning and setup.

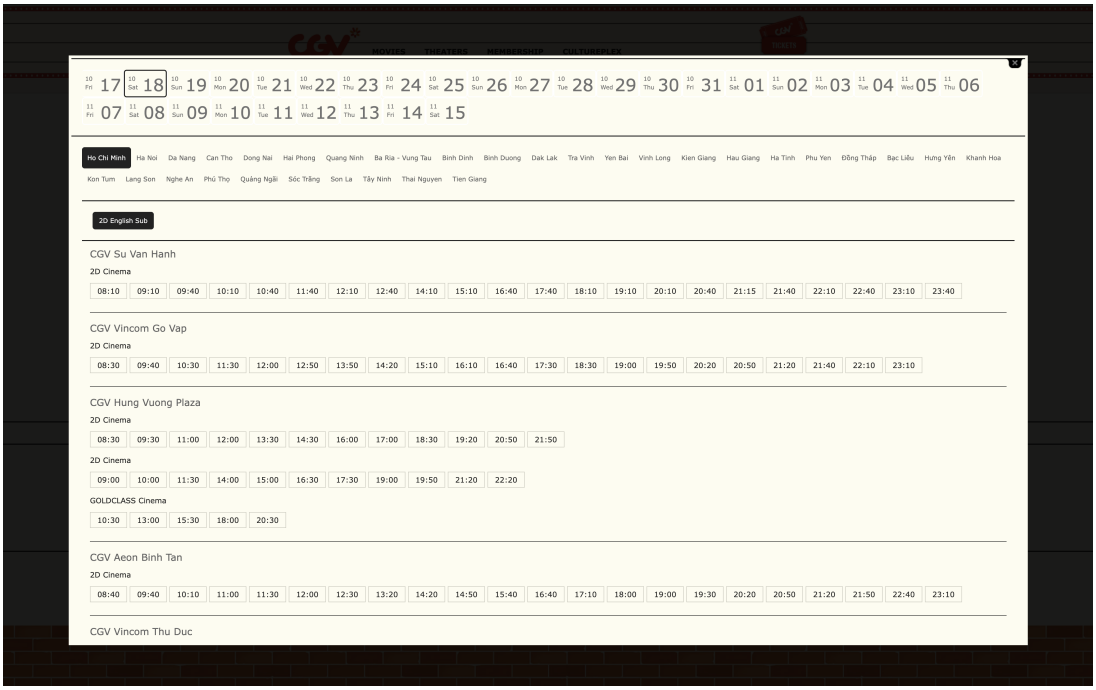

Figure 5: *Selecting cinema branch and available showtime.*

- **Choose Seats:** The seat selection interface (Figure 6) displays seat availability in real time. Once a user attempts to pay for a ticker, its seat is temporarily locked for a fixed duration (for example, payment pending time) until payment is completed to prevent double-booking conflicts.

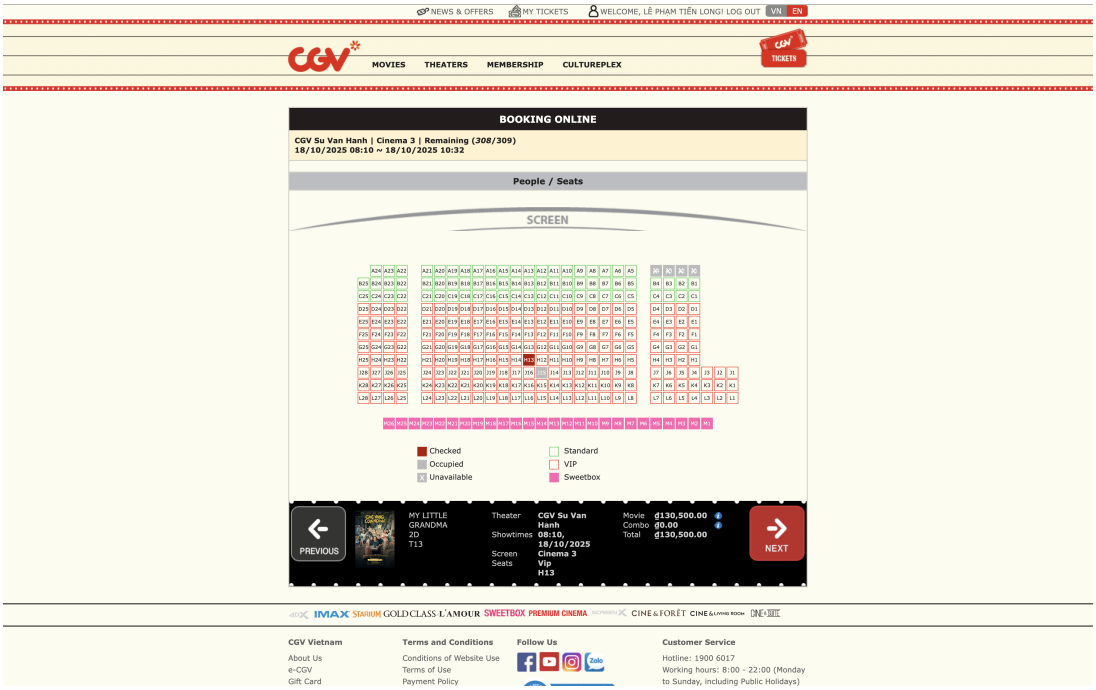

Figure 6: *Interactive seat map showing real-time seat availability.*

- **Add Food and Beverages:** After seat confirmation, users can add optional food, drinks, or combo packages (Figure 7).

Figure 7: *Adding food and beverage combos to the booking.*

- **Make Payment:** Finally, users complete the payment using debit cards, e-wallets, or reward points (Figure 8). Upon successful payment, the system records the booking creation timestamp automatically. Refund requests are only accepted at least 72 hours before the showtime.

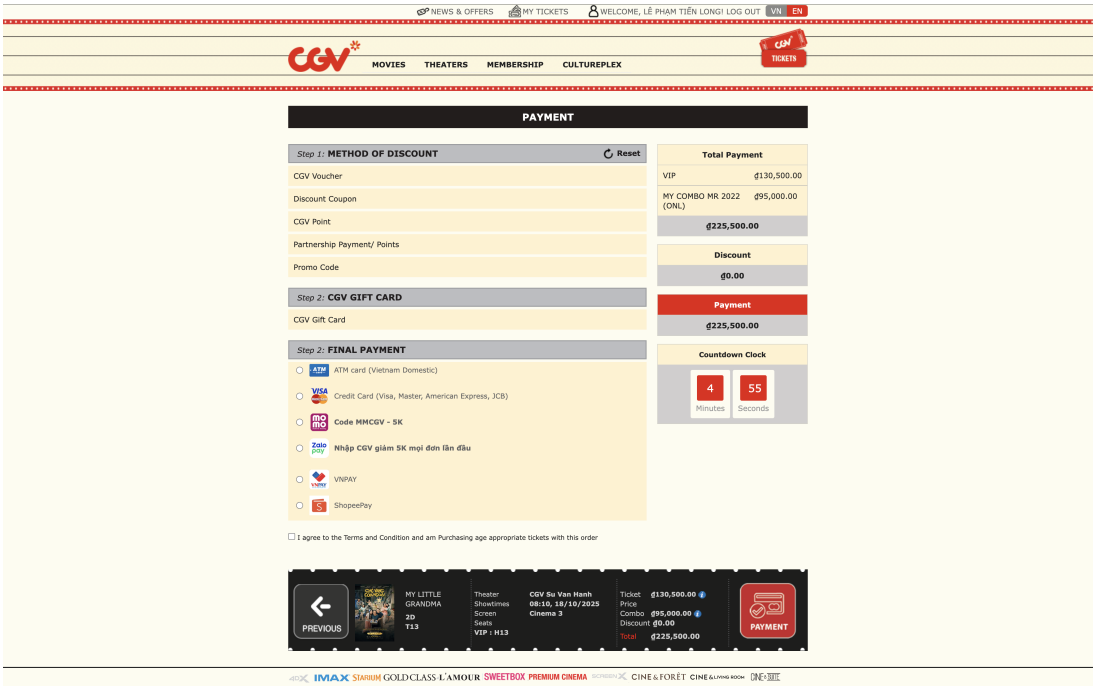

Figure 8: *Payment interface finalizing the booking transaction.*

2.1.2 Exchange Ticket and Coupon

The CGV coupon and ticket exchange system provides users with flexible options to either redeem promotional discounts or send gift vouchers to others. Unlike standard discount application, the gifting workflow involves choosing a coupon design, entering recipient information, and confirming the transaction. Each coupon is uniquely identifiable, transferable only once, and automatically linked to both the sender and the receiver for traceability.

- **View Available Coupons:** Customers can view all available coupon templates or existing coupons associated with their account (Figure 9).

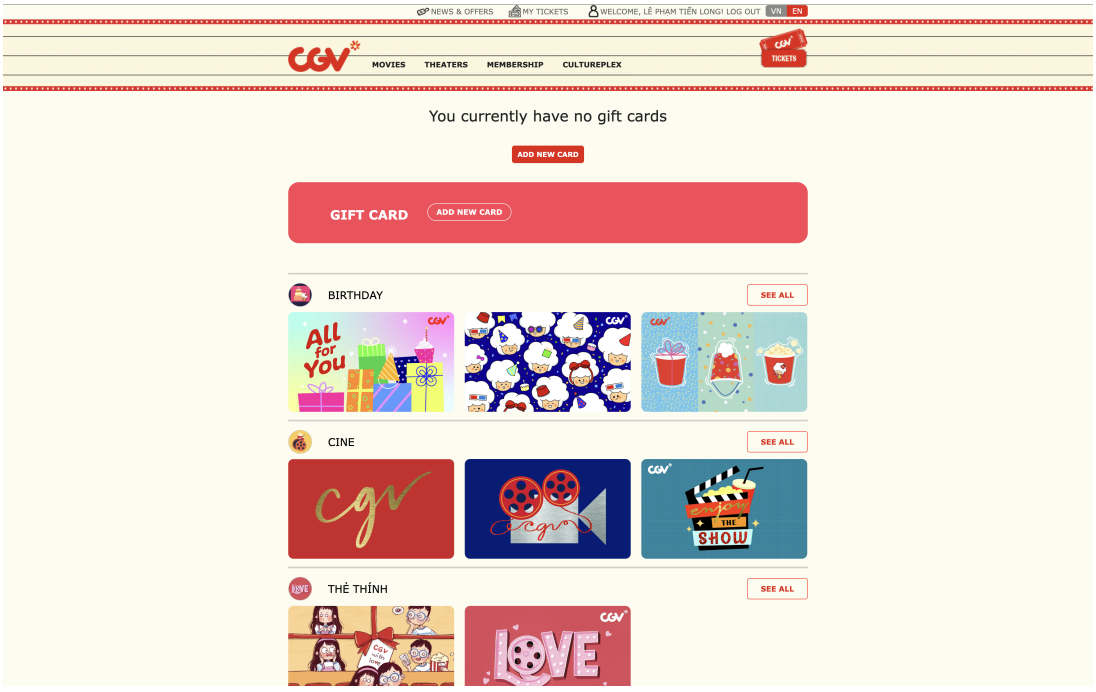

Figure 9: *Displaying available coupon templates and owned coupons in the customer account.*

- **Choose Coupon Theme:** As illustrated in Figure 10, users can choose the desired coupon design or theme before gifting. This step involves selecting a visual template that reflects the intended purpose or tone of the gift, such as festive, romantic, or minimalist styles. Each theme includes predefined color schemes, background patterns, and typography elements to enhance the presentation of the coupon.

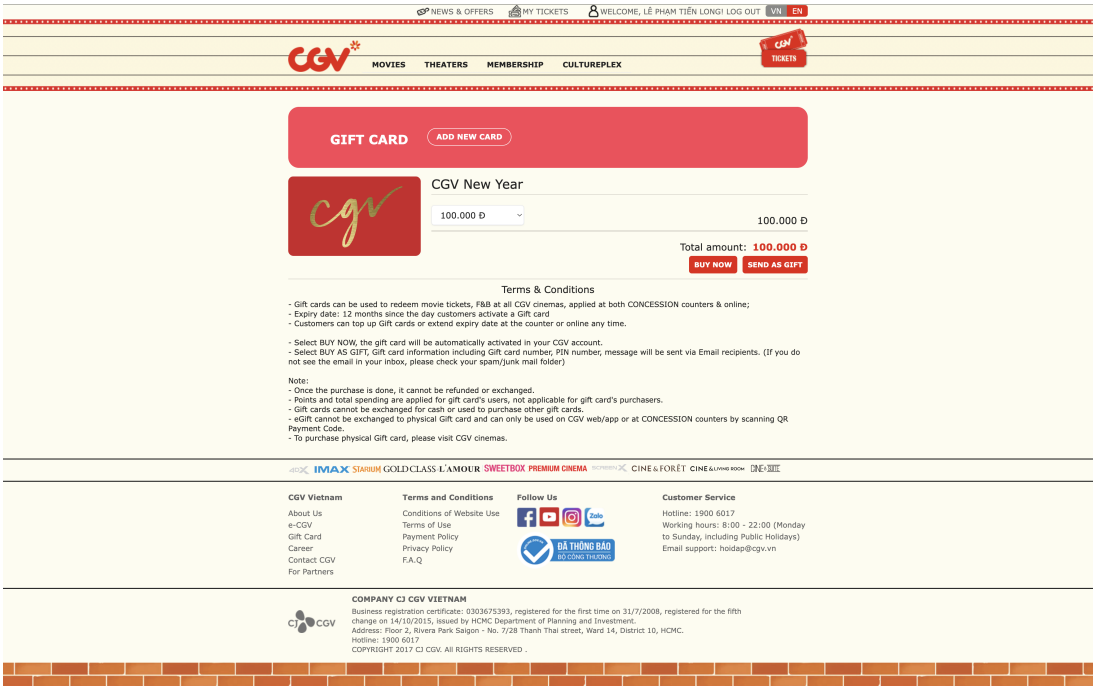

Figure 10: *Selecting a coupon theme or type for gifting.*

- **Fill in Receiver's Information:** The next step requires users to enter receiver's details such as full name, email address and message (Figure 11). These data points ensure that the coupon is associated with a valid user identity.

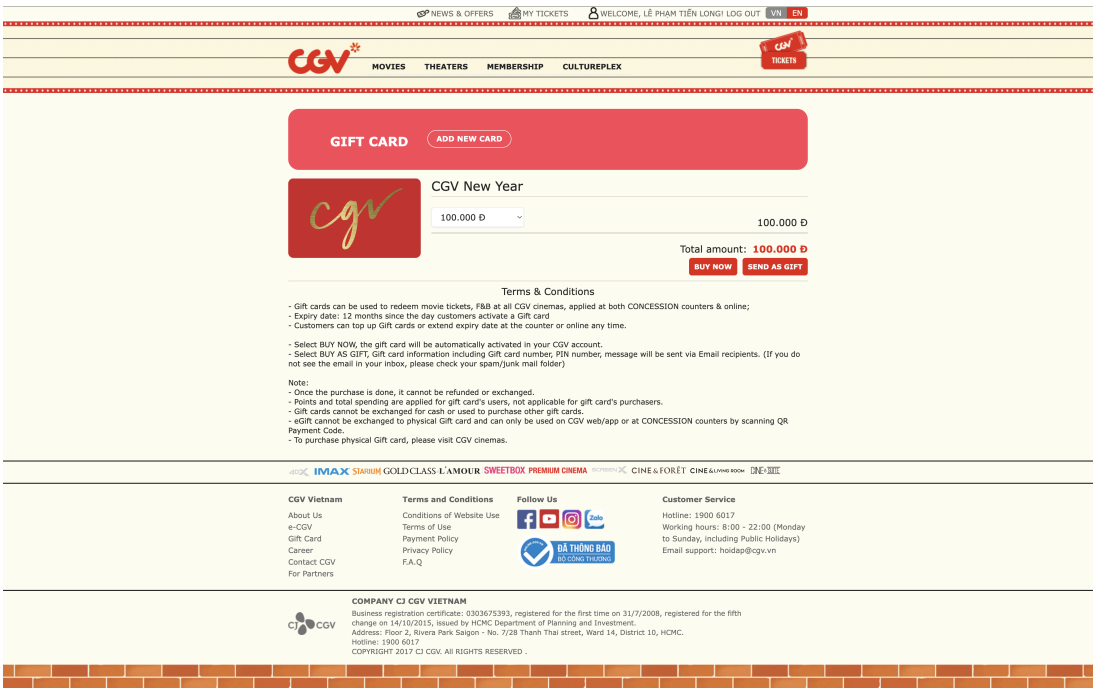

Figure 11: *Entering receiver's information to finalize coupon gifting.*

- **Confirm Details:** As shown in Figure 12, users review all entered information before sending.

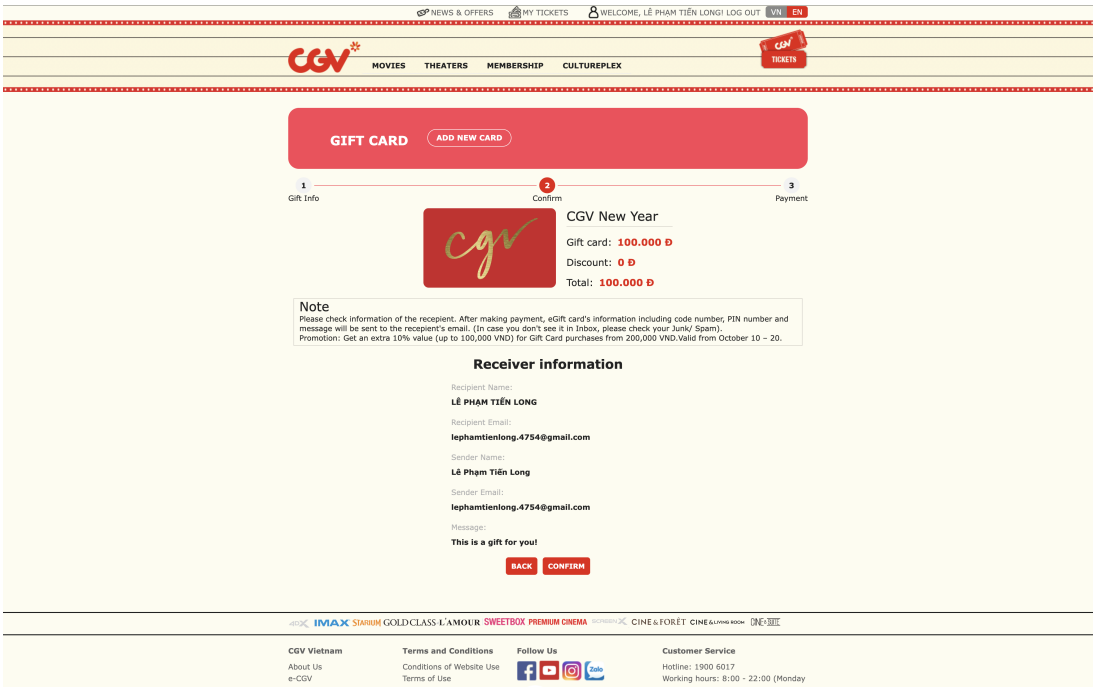

Figure 12: *Reviewing and confirming coupon details before sending the gift.*

- **Complete Payment or Transfer:** As shown in Figure 13, the system completes the coupon transfer or payment process depending on the coupon type. For paid gift coupons, payment gateways are triggered, and once successful, the system updates the receiver's wallet or account to reflect the new coupon.

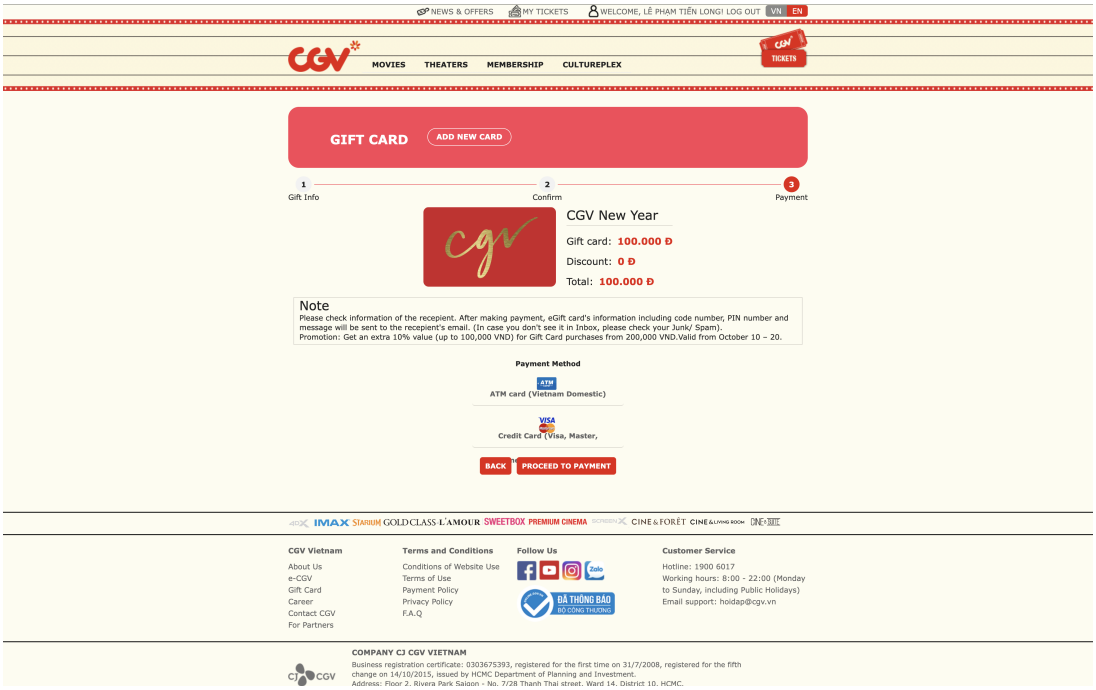

Figure 13: *Completing the payment or transfer for a gifted coupon.*

2.2 Proposed System

BKinema is an online cinema management and ticket booking system developed as a project for the Database course, based on the operation model of CGV Cinemas. The system supports customers in browsing movie information, viewing showtimes, and booking tickets online, while also providing management tools for theatre staff.
The system includes three types of users: customers, administrators, and theatre staff. Customers can view movie information, book tickets, choose seats, order food and beverages, and make online payments. They can also buy and send vouchers or bookings as gifts to other users through their accounts. Theatre staff confirm tickets at the counter, assist customers, and manage orders for food, beverages, and souvenirs. In the scope of this assignment, we design administrators as a staff but they can manage the movie catalogue, update showtimes, and handle system issues.
Main functions include movie information retrieval, which displays trailers, descriptions, duration, genre, cast, director, and ratings. The showtime schedule provides detailed information for each cinema and session, allowing filtering by time and location. The online booking module enables users to select a cinema, showtime, and seat to purchase tickets. The ordering module allows customers to choose food, drink combos, and souvenir products. The payment module supports multiple methods such as bank cards, e-wallets, and membership points. The user account module supports registration, login, booking history, and membership point tracking. The membership integration module manages loyalty cards, reward points, and member-tier promotions. The operation management module enables staff to verify e-tickets, confirm bookings, process orders, and support customers directly.

2.3 Entities, Attributes, Relationships

The cinema management system, named **BKinema**, is designed to organize and operate all activities related to movies, showtimes, bookings, customers, staff, and supplementary services. The data model is structured around a set of interconnected entities that represent the functional and operational components of the system.
The `Theater` entity represents the physical branches of the cinema chain. Each theater possesses identifying attributes including its ID, name, location, city, district, street, image, and description. A theater comprises multiple `Auditoriums`, which serve as the individual screening rooms. Each auditorium is defined by attributes such as ID, number, type, capacity, image, and description, and it belongs to one specific theater. Within each auditorium, the **Seat** entity captures information about seating arrangements. Every seat has a unique ID, row and column position, type (such as standard, VIP, or sweetbox), and its current status. Seats are contained in auditoriums and correspond to available seats in each showtime.
The `Movie` entity stores information about films, including ID, name, duration, genre, actor, director, age rating, language, subtitle, dubbing, and release date. Each movie can be issued for multiple `Showtimes`. A showtime includes attributes such as ID, start time, end time, date, and status, and it is linked to exactly one movie and one auditorium. The relationship between showtime and seat is managed by the `Showtime_seat` entity, which tracks the seat availability, type, price, and ticket ID corresponding to a specific screening.
The `User` entity represents all system accounts, encompassing both customers and staff members. User attributes include ID, first name, last name, gender, email, phone, birthday, password, address, district, and city. A user may specialize as a `Customer` or a `Staff`. Customers maintain loyalty information and transactional relationships, whereas staff members are responsible for operational activities within the theater. The `Customer` entity extends from User and includes attributes such as accumulated points and membership information. A customer can make multiple bookings, purchase food and beverage items, receive or give coupons, and accumulate points through transactions. The `Staff` entity, also derived from User, includes attributes such as role and work shift. A staff member works for a specific theater and can be assigned to several shifts depending on the schedule.
The `Membership` entity defines the loyalty program offered to customers. It consists of attributes such as tier name, minimum point requirement, and discount rate. Each customer is associated with one membership tier, which determines their privileges and point accumulation rates. Customers can also receive promotional benefits through the `Coupon` entity, characterized by coupon ID, name, type, balance, and expiration date. Coupons can be issued or applied to customer orders, and they facilitate marketing and reward operations.

Booking activities are managed by the `Booking` entity, which acts as the central hub of customer transactions. Each booking records details such as ID, creation time, live direction, status, and booking method. A booking may include movie tickets, food and beverage items, and gifts. It is created by a customer, may be processed by staff, and can generate multiple payment and refund transactions. The `FB` and `FB_Menu` entities handle the food and beverage services within the system. `FB_Menu` represents the catalog of items with attributes such as ID, name, description, capacity, price, image, and category. The `FB` entity records the actual ordered food or beverage item, including its quantity, unit price, and whether it is a gift. The relationship *Contain_item* connects `FB_Menu` and `FB`, specifying which items are included in each order and in what quantity.
Payment processing is modeled by the `Payment` entity, which stores attributes such as ID, total payment, payment method, transaction ID, amount, status, and timestamps for creation, processing, and expiration. Each payment is generated from a booking and may result in a `Refund` if necessary. The Refund entity includes refund ID, reason, amount, status, and timestamps for creation and processing. Refunds are linked back to payments through an `Apply` relationship to ensure traceability and accounting accuracy.
The `Booking` entity also supports the exchange of gifts between customers. Customers may send or receive gifts, represented as relationships `Send_gift`, `Receive_gift`, and `Give`, facilitating promotional or reward-based interactions within the system. Each booking is processed through a payment transaction and can generate or apply coupons as part of the promotional system.

2.4 Semantic Constraints

Semantic constraints are rules that ensure the data stored in the database remains consistent with real-world logic and business operations of the BKinema system. These constraints cannot be expressed solely through the ERD model and must instead be enforced at the logical or application level. The following constraints are categorized according to their corresponding entities and relationships.

User Constraints

- Each `User` must have a unique and valid email address to prevent duplicate accounts.
- The `password_hash` field must not be null and must comply with security standards such as minimum length and hashing policy.
- The `birthday` attribute must ensure that a user's age is greater than or equal to 13 years old for account creation.
- A `Customer`'s accumulated points must always be greater than or equal to 0.
- Each `Customer` can only belong to only one membership tier for which they meet or exceed the required minimum points.
- A `Customer` cannot simultaneously hold two identical membership tiers.
- Each `Staff` member must be assigned a valid work shift and role; null assignments are not allowed.

Movie Constraints

- The `duration_minutes` of a movie must be greater than 0.
- The `release_date` of a movie must be earlier than any associated showtime's start time.
- The `age_rating` must follow a predefined set (e.g., P, K, T13, T16, T18, C) [5].
- The combination of movie name and release date must be unique to prevent duplicates across different versions.

Theater and Auditorium Constraints

- Each `Auditorium` must belong to exactly one `Theater`.
- The `capacity` of an auditorium must be greater than 0.
- The total number of `Seats` in an auditorium must not exceed its capacity.
- Each `Seat` must have a unique combination of row and column identifiers within its auditorium.
- No two seats in the same auditorium may share identical coordinates (row and column).
- Each seat must be categorized into one type of seat.
- Seat prices must be greater than or equal to 0.

Showtime and Seat Constraints

- A `Showtime` must have a start time earlier than its end time.
- Two showtimes in the same auditorium must not overlap in time.
- The time interval between two consecutive showtimes in the same auditorium must be at least 15 minutes to allow for cleaning and audience transition.
- Each `Showtime_seat` must correspond to a valid `Seat` and a valid `Showtime`.
- A seat cannot be booked by more than one customer for the same showtime.
- The `status` of a showtime seat must belong to the predefined domain {available, locked, booked}.

Booking and Ticket Constraints

- Each `Booking` must be associated with exactly one `Customer`.
- The number of tickets in one booking must be greater than or equal to 0 and less than or equal to the total available seats for that showtime.
- A booking cannot contain tickets for multiple showtimes or different seat types.
- A customer cannot book two tickets for the same seat in the same showtime.
- Ticket prices must be greater than or equal to 0.
- The booking status must belong to the domain {pending, confirmed, cancelled}.
- When a booking is cancelled, all associated tickets and payments must be automatically invalidated.
- The system must prevent leaving an empty seat between two booked seats in the same row (no single-seat gap rule).

Booking and F&B Constraints

- Each `F&B` item must be associated with exactly one valid `Booking` record.
- A booking can include zero or more `F&B` items.
- Each `F&B` item's price must be greater than or equal to 0.
- The quantity of each F&B item in a booking must be greater than or equal to 1.
- A booking cannot contain the same F&B item multiple times with different price records.
- If a booking is cancelled, all associated F&B items must also be voided and excluded from inventory accounting.
- Discounts from memberships or coupons must be applied consistently to both ticket and F&B totals under the same booking.
- If an F&B item becomes unavailable (out of stock) before payment confirmation, the payment and booking will be cancelled.

Payment and Refund Constraints

- Each `Payment` must correspond to exactly one confirmed booking.
- A payment record must be created before its expiration time.
- The `amount` and total value must match the sum of ticket and F&B prices minus discounts.
- The `status` attribute of a payment must belong to the domain {successful, pending, failed}.
- A `Refund` can only occur if the associated booking is cancelled and the payment status is successful.
- The refund amount must be less than or equal to the original payment amount.

Coupon and Gift Constraints

- A `Coupon` cannot be applied after its expiration date.
- The balance of a coupon must be greater than or equal to 0.
- A coupon marked as a gift cannot be reused after being redeemed.
- A `Customer` cannot send a gift coupon to themselves.
- Each gift record must have distinct sender and receiver IDs.

General Business Rules

- Customers below the movie's `age_rating` are not permitted to book tickets for that movie.
- Bookings cannot be created for showtimes that have already ended.
- Each booking must record the creation timestamp automatically upon submission.
- All monetary attributes (price, amount, balance) must maintain precision up to two decimal places.
- Refund requests must be submitted at least 72 hours before the showtime's scheduled start to be eligible for approval.
- A refund cannot be processed after the showtime has started, regardless of booking status.
- Coupons and membership discounts cannot be applied simultaneously within the same booking unless explicitly allowed by policy.
- A coupon may only be used once per booking and cannot be reused in future transactions.
- Membership discounts are calculated based on the customer's membership tier at the time of booking creation, not at payment confirmation.
- Gift bookings must include a valid recipient record and cannot be self-gifted.
- If a showtime is cancelled by the theater, all affected bookings must be automatically refunded in full.
- A `Coupon` can only be gifted once.
- A `Booking` can only be transferred once.
- Online ticket booking must be completed at least 15 minutes before the corresponding showtime's `start_time`. After this threshold, the system will disable online booking for that showtime.
- Seats are temporarily locked only when a customer reaches the payment stage. The system prevents other users from selecting those seats during the payment process.
- If the payment process is cancelled, interrupted, or times out, all temporarily locked seats associated with that transaction are immediately released and reverted to the `available` state.

- A seat is considered successfully booked only after full payment confirmation. Any payment failure or cancellation automatically invalidates the corresponding booking record.
- In case of simultaneous booking attempts for the same seat, the system follows a *first-payment-confirmed* policy.

3 Entity–Relationship Diagram

The ERD of the BKinema system provides a conceptual representation of how data entities are structured and interconnected within the database. It serves as the foundation for designing the relational schema by illustrating the entities, their attributes, and the various relationships among them, including one-to-one, one-to-many, and many-to-many associations.
The ERD captures the main components of the cinema management system, such as `User`, `Customer`, `Staff`, `Movie`, `Theater`, `Auditorium`, `Seat`, `Showtime`, and `Booking`, along with supporting entities like `Payment`, `Refund`, `Coupon`, and `Membership`. Weak entities, such as `Seat` and `Showtime_Seat`, are represented with identifying relationships that depend on their parent entities. Moreover, specialization is applied to model the inheritance between `User`, `Customer`, and `Staff`. Figure 14 illustrates the complete ERD of the BKinema database.

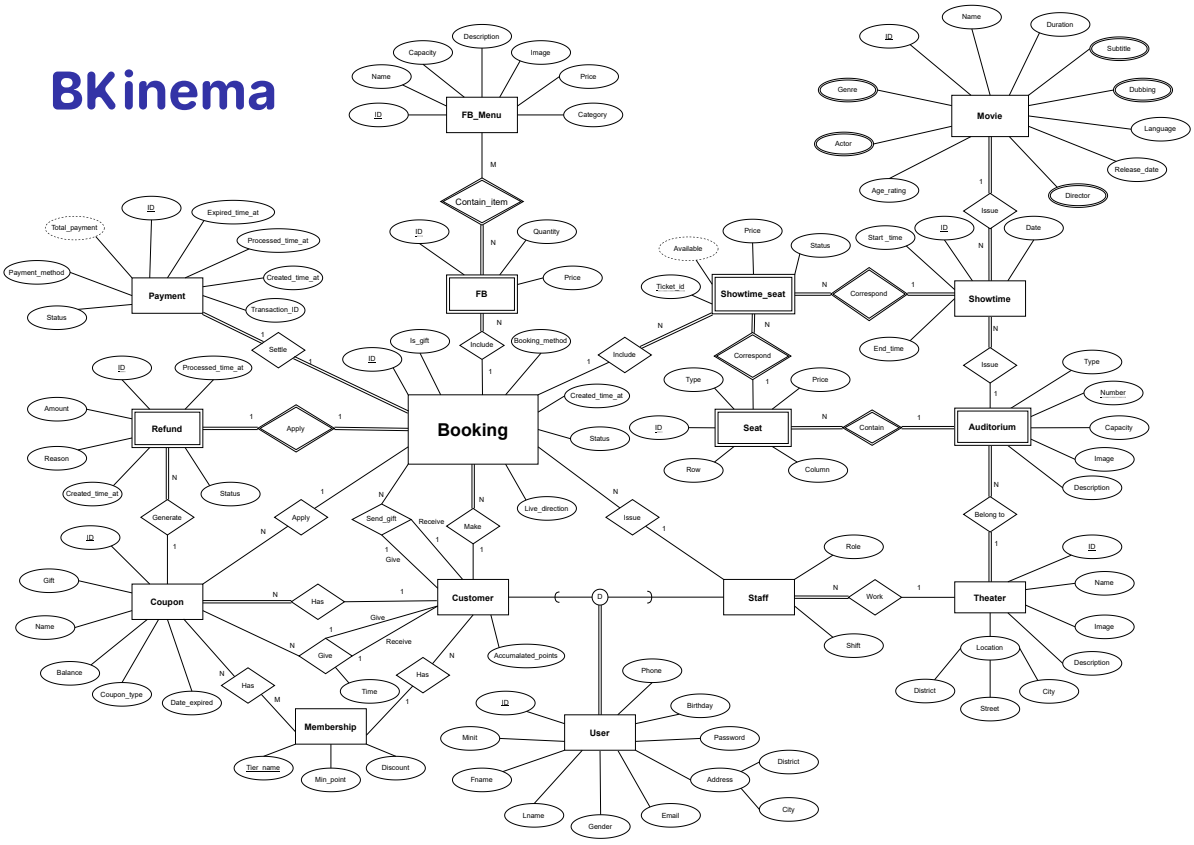

Figure 14: *ERD of the BKinema database.*

4 Database Schema

Entity Types

- `User`(`ID`, `Fname`, `Minit`, `Lname`, `Birthday`, `Gender`, `Email`, `Password`, `District`, `City`)
  - Primary key: `ID`
  - Secondary (unique, not null) key: `Email`

- Customer(User_ID, Accumulated_points, Membership_Name)
  - Primary key: User_ID
  - Foreign keys: User_ID → User.ID, Membership_Name → Membership.Tier_name
- Staff(User_ID, Shift, Role)
  - Primary key: User_ID
  - Foreign key: User_ID → User.ID
- Movie(ID, Name, Duration, Language, Release_date, Age_rating)
  - Primary key: ID
- Theater(ID, Name, Street, District, City, Image, Description, Staff_ID)
  - Primary key: ID
  - Foreign key: Staff_ID → Staff.User_ID
  - Not null: Name, Street, District, City
- Auditorium(Number, Theater_ID, Type, Capacity, Image, Description)
  - Primary key: (Number, Theater_ID)
  - Foreign key: Theater_ID → Theater.ID
  - Default: Type = Normal
  - Not null: Type, Capacity
- Seat(ID, Au_Number, Au_Theater_ID, Row, Column, Type, Price)
  - Primary key: (ID, Au_Number, Au_Theater_ID)
  - Foreign key: (Au_Number, Au_Theater_ID) → (Auditorium.Number, Auditorium.Theater_ID)
  - Note: Weak entity dependent on Auditorium
- Showtime(ID, Date, Start_time, End_time, Movie_ID, Au_Number, Au_Theater_ID)
  - Primary key: ID
  - Foreign keys: Movie_ID → Movie.ID, (Au_Number, Au_Theater_ID) → (Auditorium.Number, Auditorium.Theater_ID)
  - Not null: Date, Start_time, End_time
- Showtime_seat(Ticket_ID, ST_ID, Seat_ID, Seat_Au_Number, Seat_Au_Theater_ID, Status, Price, Booking_ID)
  - Primary key: (Ticket_ID, ST_ID, Seat_ID, Seat_Au_Number, Seat_Au_Theater_ID)
  - Secondary (unique, not null) key: (Showtime_ID, Seat_ID)
  - Foreign keys: ST_ID → Showtime.ID, (Seat_ID, Seat_Au_Number, Seat_Au_Theater_ID) → (Seat.ID, Seat.Au_Number, Seat.Au_Theater_ID), Booking_ID → Booking.ID
  - Default: Status = Available
  - Note: Weak entity dependent on Showtime and Seat
- Booking(ID, Created_at, Status, Booking_method, Is_gift, Live_direction, Customer_ID, Staff_ID)
  - Primary key: ID
  - Foreign keys: Customer_ID → Customer.User_ID, Staff_ID → Staff.User_ID
  - Not null: Booking_method, Status

- Payment(ID, Payment_method, Status, Created_at, Transaction_ID, Expired_at, Duration, Booking_ID)
  - Primary key: ID
  - Secondary (unique, not null) key: Transaction_ID
  - Foreign key: Booking_ID → Booking.ID
- Refund(ID, Booking_ID, Amount, Reason, Status, Created_at, Processed_at, Coupon_ID)
  - Primary key: (ID, Booking_ID)
  - Foreign keys: Booking_ID → Booking.ID, Coupon_ID → Coupon.ID
  - Note: Weak entity dependent on Booking
- Coupon(ID, Name, Gift, Balance, Coupon_type, Date_expired, Booking_ID, Customer_ID)
  - Primary key: ID
  - Foreign keys: Booking_ID → Booking.ID, Customer_ID → Customer.User_ID
- Membership(Tier_name, Min_point, Discount)
  - Primary key: Tier_name
  - Unique: Min_point, Discount
- FB_Menu(ID, Name, Description, Image, Price, Category, Capacity)
  - Primary key: ID
- FB(ID, Booking_ID, Quantity, Price)
  - Primary key: (ID, Booking_ID)
  - Foreign key: Booking_ID → Booking.ID
  - Note: Weak entity dependent on Booking and FB_Menu
- Actor(ID, Name, Gender, Image)
  - Primary key: (ID)
  - Foreign key: ID → Movie.ID
- Genre(ID, Name)
  - Primary key: (ID)
  - Foreign key: ID → Movie.ID
- Subtitle(ID, Language)
  - Primary key: (ID)
  - Foreign key: ID → Movie.ID
- Dubbing(ID, Language)
  - Primary key: (ID)
  - Foreign key: ID → Movie.ID
- Director(ID, Name, Gender, Image)
  - Primary key: (ID)
  - Foreign key: ID → Movie.ID

Tenary Relationships (1:1:N)

- Send_gift(Booking_ID, Sender_ID, Receiver_ID)

  - Primary key: `Booking_ID`
  - Foreign keys: `Booking_ID` → `Booking.ID`, `Sender_ID` → `User.ID`, `Receiver_ID` → `User.ID`
  - Not null: `Sender_ID`, `Receiver_ID`

- `Give`(`Coupon_ID`, `Sender_ID`, `Receiver_ID`)
  - Primary key: `Coupon_ID`
  - Foreign keys: `Coupon_ID` → `Coupon.ID`, `Sender_ID` → `User.ID`, `Receiver_ID` → `User.ID`
  - Not null: `Sender_ID`, `Receiver_ID`

N:N Relationships

- `Contain_item`(`FB_Menu_ID`, `FB_ID`, `FB_Booking_ID`, `Quantity`)
  - Primary key: (`FB_Menu_ID`, `FB_ID`)
  - Foreign keys: `FB_Menu_ID` → `FB_Menu.ID`, (`FB_ID`, `FB_Booking_ID`) → (`FB.ID`, `FB.Booking_ID`)
  - Note: Represents N:N relationship between `FB_Menu` and `FB`
- `Has`(`Coupon_ID`, `Membership_Tier_Name`)
  - Primary key: (`Coupon_ID`, `Membership_Tier_Name`)
  - Foreign keys: `Coupon_ID` → `Coupon.ID`, `Membership_Tier_Name` → `Membership.Tier_name`

Specialization / Generalization

- Relation: *is-a* relationship among `User`, `Customer`, and `Staff`
- Description: `Customer` is a subtype of `User`, and `Staff` is also a subtype of `User`. This represents a specialization where `User` is the superclass and `Customer`/`Staff` are subclasses.
- Constraint: Disjoint and partial specialization (a `User` cannot be both `Customer` and `Staff` simultaneously, and not all `Users` are required to be either)

Based on the detailed mapping and ERD presented above, the structure of the BKinema database schema has been designed as shown in Figure 15.

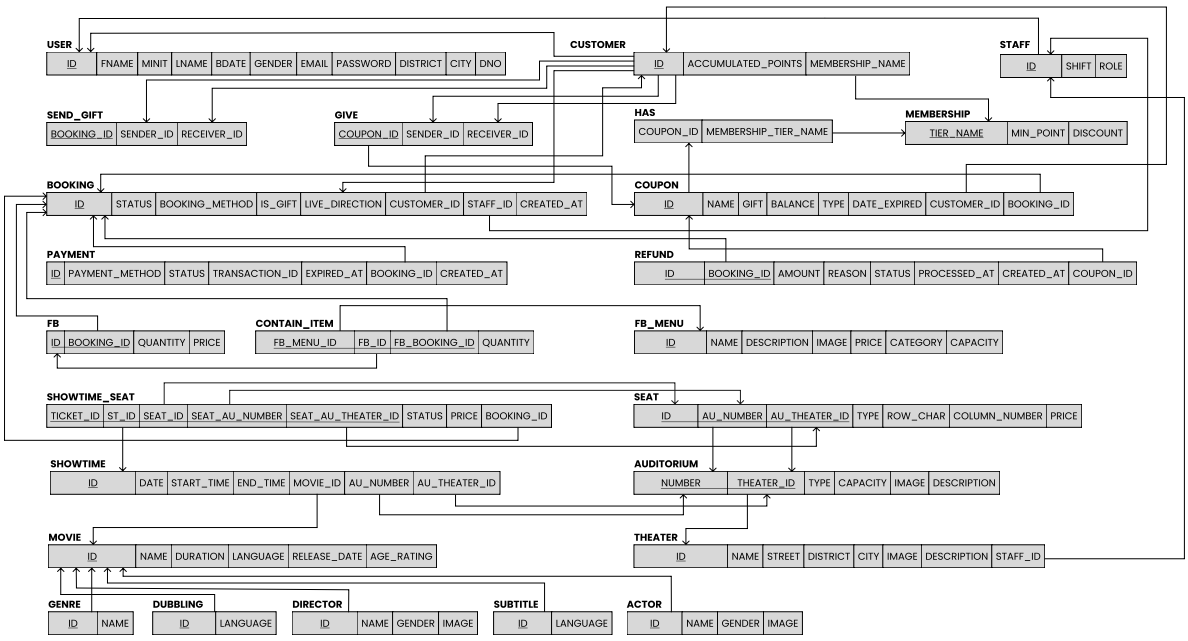

Figure 15: *Schema of the BKinema database.*

Based on the mapping schema (Figure 15) and ERD (Figure 14), all binary relationships in the BKinema database are systematically transformed into relational tables following standard database normalization principles. For the **1:N relationships**, the relationship between `Theater` and `Auditorium` is implemented by adding `Theater_ID` as a foreign key in the `Auditorium` table, indicating that each auditorium belongs to one theater, but a theater can contain many auditoriums. Similarly, the `Seat` table contains the composite foreign key (`Au_Number`, `Au_Theater_ID`) referencing the `Auditorium` table, reflecting that multiple seats exist within a single auditorium.
For the **1:1 relationships**, the relationship between `User` and its subtypes `Customer` and `Staff`, each subtype table (`Customer`, `Staff`) uses `User_ID` as both a primary and foreign key referencing `User(ID)`. This maintains the one-to-one correspondence and ensures specialization integrity (each customer or staff corresponds to exactly one user record). Another one is the relationship between `Booking` and `Payment`, where the `Booking_ID` foreign key in `Payment` enforces that each payment is uniquely linked to one booking. This design maintains data consistency and supports transaction traceability. Similarly, the `Refund` table also references `Booking(ID)` through a composite primary key (`ID`, `Booking_ID`), reflecting a dependent weak entity relationship.

5 Conclusion

This report explores how a cinema booking system is constructed. Through the construction of the ERD and the transformation into a relational schema, the study has modeled all critical components of a modern cinema system, including user management, movie scheduling, ticket booking, payment processing, and loyalty services.
The proposed implementation of the BKinema system adopts a web-based architecture. The backend will be developed using **Express.js** with RESTful APIs to handle business logic and interact with the database. The frontend interface will be implemented using **React.js**, providing a user experience across web and mobile platforms. The system's data layer will be managed by **Supabase PostgreSQL**. Finally, **Vercel** will serve as the hosting and deployment platform.

Acknowledgement

We would like to express our sincere gratitude to Mr. Le Duc Hoang Nam for valuable guidance and support throughout this study.

References

[1] SkyQuest, *Online Movie Ticketing Services Market Size, Share, and Growth Analysis (2025–2032)*, `https://www.skyquestt.com/report/online-movie-ticketing-services-market`, Accessed: Oct. 2025, Jan. 2025.

[2] VietNamNews, *The total box office revenue in Viet Nam for 2024 is the highest-ever*, `https://e.vnexpress.net/news/life/culture/vietnam-box-office-record-2024-4744829.html`, Accessed: Oct. 2025, Jan. 2025.

[3] B&Company, *Vietnam Cinema Market and Room for Growth*, `https://b-company.jp/vietnam-cinema-market-and-room-for-growth`, Accessed: Oct. 2025, Oct. 2024.

[4] DataReportal, *Digital 2025: Vietnam*, `https://datareportal.com/reports/digital-2025-vietnam`, Accessed: Oct. 2025, Mar. 2025.

[5] Ministry of Culture, Sports and Tourism, *Circular No. 05/2023/TT-BVHTTDL on Regulations on Film Classification Criteria and Implementation of Film Rating Display and Warnings*, `https://bvhttdl.gov.vn/van-ban-quan-ly/110688.htm`, Accessed: October 2025, Apr. 2023.