A cinema system

Group 6: Weijie Guan, Yining Wang

Business problem definition:

With the development of 3D and iMax technology, the experience of watching films in the cinema becomes increasingly better than that at home. People tend to go out for entertainment nowadays. As the number of customers grows, traditional management may require more human resources and financial resources. A management method with higher efficiency and confidentiality will help the cinema to better improve its operation. We designed a ticketing database to meet these requirements.

Requirements:

The ticketing database for a cinema should meet the following requirements:

1. It has the basic functions including playing the films, selling tickets, checking tickets, dealing with useless data.
2. It can be applied to the snack corner, either.
3. It can deal with the expired tickets.
4. A customer can choose whether to buy a ticket for one time, or to join the vip club which has more discounts.

We suppose that if a customer want to join the VIP club, he must prepay for a VIP card. Each consumption in the cinema using the VIP card will add a credit of 10% of the bill to his account. He can use the credit to buy goods in the snack corner or buy film tickets.

The ticketing database is made up of 3 parts, ticket selling, checking and maintaining. For the maintaining system, there are two factors including staff and VIP members.

The authority of a staff:

1. Adding, searching, editing and deleting a VIP member's information in the system.
2. Adding or deducting credits and balance in a customer's VIP account.
3. Looking up the information of a film to be played and finding out its hall number.
4. Looking up the tickets left and the tickets sold together with the seats they match.
5. Selling tickets.
6. Adding, deleting and editing the information of goods.
7. Stock management.
8. Selling goods.
9. Checking the verification code for a ticket.
10. Checking the date and time for a ticket.

Information for a VIP member includes: card number, first name, last name, address, balance in the card, credits in the card

Information for a film includes: film code, name of the film, director, leading role, nationality, release date

Information for a seat: seat number, whether it is occupied, room number

Information for a room: room number, type, capacity

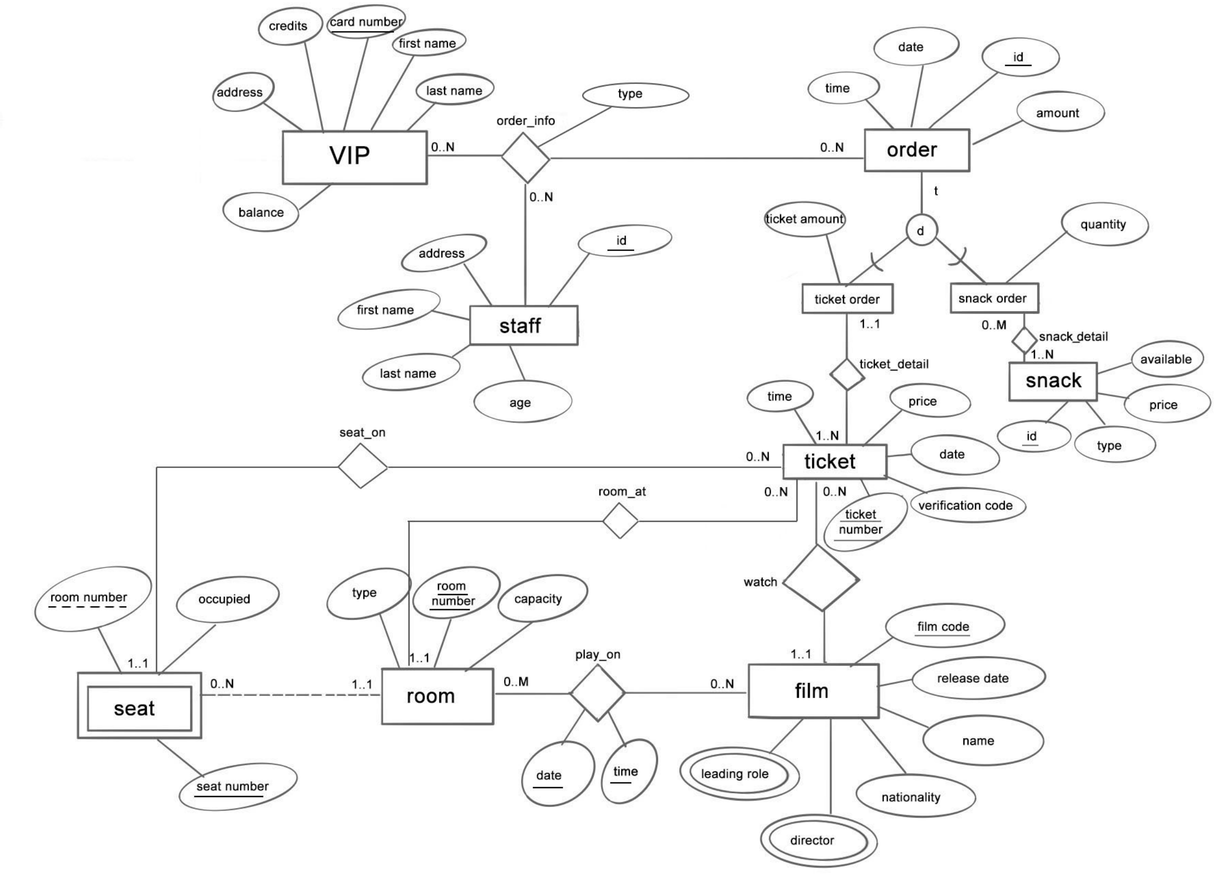
Information for a snack: availability, price, type, id

Information for a ticket: verification code, film number, name of the film, ticket number, seat number, room number, price, date, time

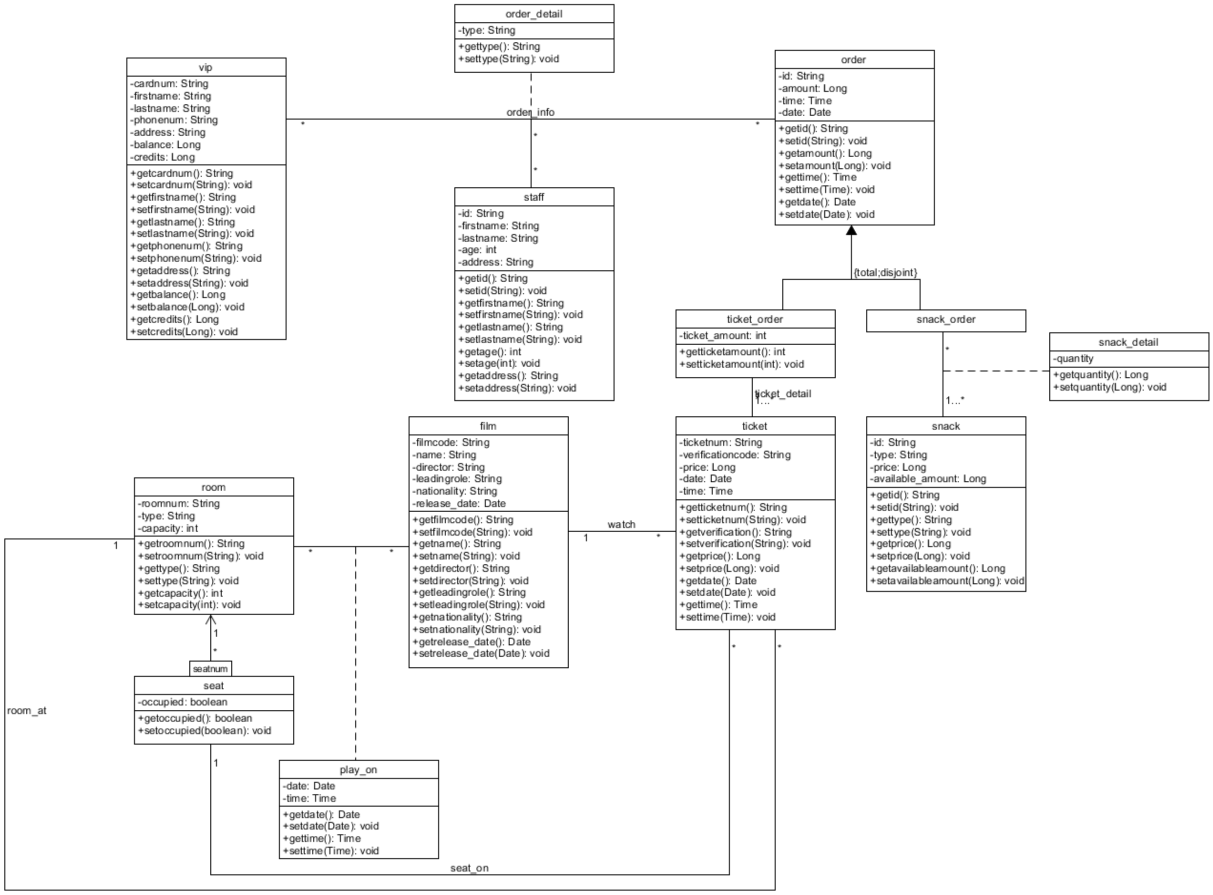
Information for the staff: age, id, address, first name, last name

Each bill is recorded as an order, the information for an order includes: time, date, id, amount

EER Model



UML Class Diagram



Relation Model

Vip(cardnum, firstname, lastname, phonenum, address, balance, credits)

Room(roomnum, type, capacity)

Seat(seatnum, occupied, roomnum)

-roomnum: foreign key refers to roomnum in Room, NULL value not allowed

Film(filmcode, name, director, leadingrole, nationality, release_date)

Ticket(ticketnum, verificationcode, price, date, time, filmnum, roomnum, seatnum, order_id)

-filmnum: foreign key refers to filmcode in Film, NULL value not allowed

-roomnum: foreign key refers to roomnum in Room, NULL value not allowed

-seatnum: foreign key refers to seatnum in Seat, NULL value not allowed

-order_id: foreign key refers to id in Ticket_order, NULL value not allowed

Snack(id, type, price, available_amount)

Order(id, amount, time, date)

Snack_order(id)

Ticket_order(id, ticket_amount)

Play_on(filmcode, roomnum, date, time)

-filmcode: foreign key refers to filmcode in Film, NULL value not allowed

-roomnum: foreign key refers to roomnum in Room, NULL value not allowed

Staff(id, firstname, lastname, age, address)

Order_info(vip_id, staff_id, order_id, type)

-vip_id: foreign key refers to cardnum in Vip, NULL value allowed

-staff_id: foreign key refers to id in Staff, NULL value not allowed

-order_id: foreign key refers to id in Order, NULL value not allowed

Snack_detail(snack_id, order_id, quantity)

-snack_id: foreign key refers to id in Snack, NULL value not allowed

-order_id: foreign key refers to id in Snack_order, NULL value not allowed