Theater Ticketing Software

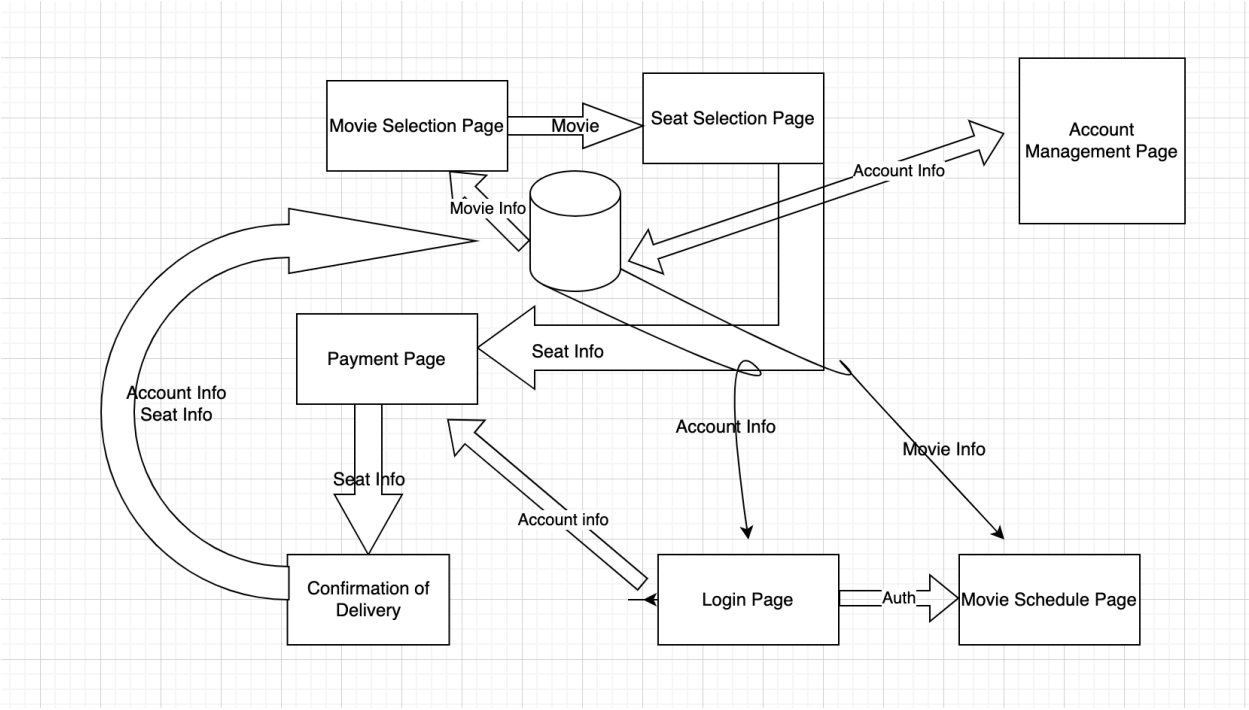
Prepared by: Cameron Croom, Gilbert Reyno, Victoria Nguyen, Grace Peebles, Mark Langell

February 24, 2023

System Description:

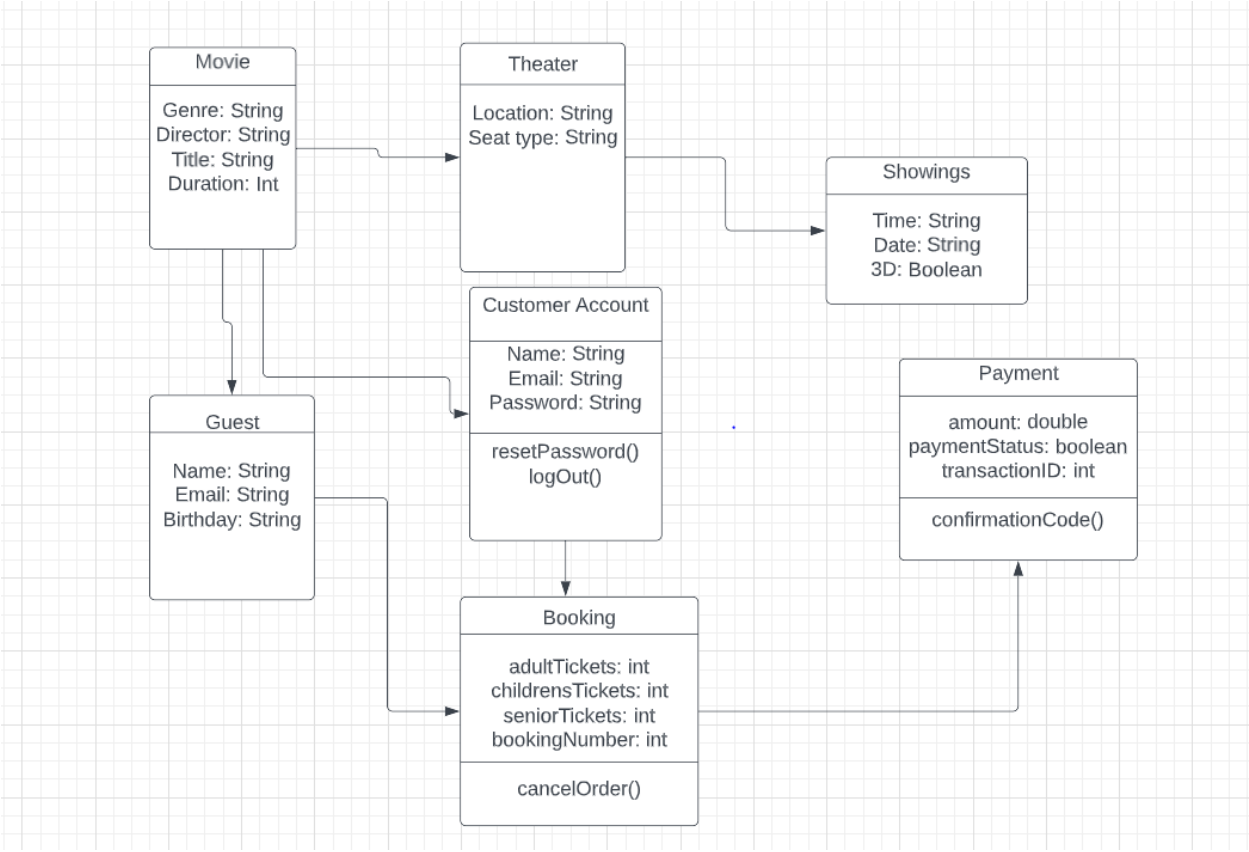
The purpose of this system is to be able to buy movie tickets from our designed software. This software stores and contains the necessary information and components of a movie theater such as the theaters, their showings, and the ticket prices and booking. This software will also have the ability to collect user input such as how many tickets they want to purchase as well as the status of the guests (child, normal, or senior) and change certain outputs depending on the input. The system will also be able to search for nearby theaters for ticket availability and provide payment confirmation. The system has language support for English, Spanish, and Swedish. Users can register loyalty accounts that store personal information, payment information, purchase history, loyalty points. The system will be able to handle up to 10 million concurrent users and the tickets must be unique and non replicable.

Architectural Diagram:



In the Software architecture diagram, each connector has a label which points to the next component that determines what the system's next step is. For example, when on the Movie Selection Page, the user selects a movie and is then directed to the Seat Selection Page.

UML Class Diagram:



Movie: this class stores the attributes of a movie so that it may be identifiable to other classes and assigned to guests.

- **Genre:** a string attribute that contains the general category of the movie.
- **Director:** a string attribute that contains the person who directed the movie.
- **Title:** a string attribute that contains the title of the movie, this is most commonly used to identify a movie.
- **Duration:** a string attribute that contains the time duration of the movie.

Guest: this class stores the attributes of a single guest.

- **Name:** a string attribute that contains the individual name of the guest.
- **Email:** a string attribute that contains the email address of the guest.
- **Birthday:** a string attribute that displays the yearly birthday of the guest.

Theater: this class stores the attributes of a theater, a place where movies are displayed and guests watch them.

- **Location:** a string attribute that contains the physical location of the theater.

- **Seat type (aka standard or luxury):** a string attribute that contains the type of seats in the theater.

Showings: this class stores the attributes of the showings of movies, or when the movies are played.
- **Time:** a string attribute that contains the time that a showing takes place.
- **Date:** a string attribute that contains the date that a showing takes place.
- **3D:** a boolean attribute that identifies whether the showing is in 3D or not.

Customer Account: this class stores the attributes of an account tied to a customer.
- **Name:** a string attribute that contains the individual name of the owner of the account.
- **Email:** a string attribute that contains the email address of the owner of the account.
- **Password:** a string attribute that contains the "password" or key that identifies the owner of the account.
    - **resetPassword():** an operation that allows the user to "reset" or change the password attribute by stating their old password to confirm they are the owners of the account, and then passing their new password.
    - **logOut():** an operation that allows the user to "log out" or stop actively using the account. that they are tied to by stating that they want to log out.

Booking: this class stores the booking of one or multiple tickets to a movie.
- **adultTickets:** an integer value that contains the number of tickets that belong to adults.
- **childrensTickets:** an integer value that contains the number of tickets that belong to children.
- **seniorTickets:** an integer value that contains the number of tickets that belong to seniors.
- **bookingNumber:** an integer value that contains the identification number of the booking.
    - **cancelOrder():** an operation that allows the order of the booking to be canceled.

Payment: this class stores the attributes of a payment for a booking to a movie.
- Amount: a double value that stores the dollar amount that will be owed in the payment.
- paymentStatus: a boolean attribute that identifies whether or not the payment has been completed.
- transactionID: an integer value that contains the identification number of the transaction.
    - confirmationCode() an operation that confirms to the paying and receiving parties that the payment has been completed.

Development plan and timeline

- Partitioning of tasks:
    - Victoria will be working on making the system user-friendly.
    - Cameron will work on the booking and payment system.
    - Grace will be working on creating customer and guest accounts.
    - Gilbert will be working on the movies and the theaters.
    - Mark is going to work in tandem with Gilbert, but will focus on showings as well

- Team member responsibilities
    - Each member will need to document their code and progress, as well as communicate with the other members on the team in order to make the needed connections possible.
    - The user friendliness of the system is mostly focused on UI, which is universal on all categories.
    - The booking and payment system must keep track of the prices of admission as well as transactions between the theater and the customer.
    - The customer and guest accounts system should encourage customers to create and maintain an account, and should work closely with the member in charge of user friendliness.
    - The movies and theaters system should track all of the current movies and theaters and their attributes. It should also work closely with the showings system.
    - The showings system should note all of the showings that take place in the theater, and should work closely with the movies and theaters system as that is where the showings will take place.