

Software Design Specification

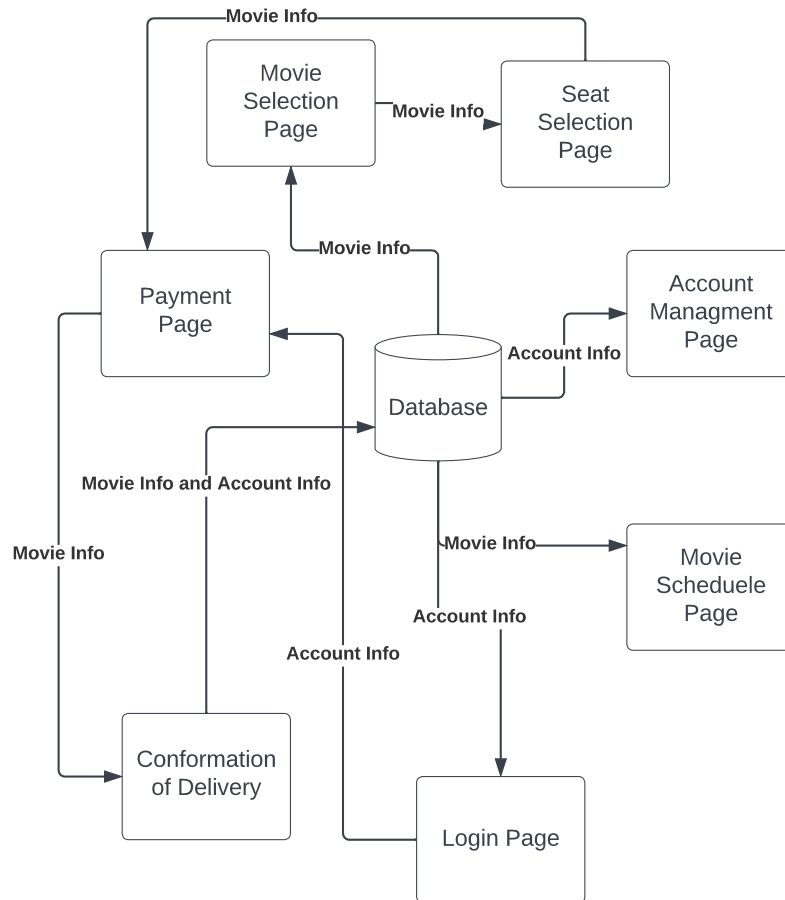
By: Luis Suarez & Hailun Xu

System Description

The system to be constructed will have the main function of being a ticketing service for a local San Diego cinema chain. It will also host a secondary function of being a customer feedback system, a user profile service, and an AMC subscription service.

Software Architecture Overview

The architecture diagram of all major components is the following:



<showtimes, locations, seats, prices> = movie info

<username, password, authorization loyalty points, payment information, purchase history> = account info.

The architectural diagram will be the following:

Database: Will hold all the variables set by the users that, broadly, make up the account information, movie information, seat information

Payment Page: Will be the interface for payment of a movie ticket(s). It will take in seat information and account information that will complete the payment transaction---. The Payment Page will then also send the seat information for there to the Conformation of Delivery interface.

Conformation of Delivery: Will be the interface for confirming the (electronic) delivery of tickets to the user. It will take in the seat information from the Payment Page to display the seat number for the user's ticket. Then it will send the seat information and account information back into the database for storage

Login Page: Will be the interface for logging into a user's cinema account. It will take in the account information from the database to compare the input put in by the user to see if the username and corresponding password are valid. Then, after a successful login, the Login Page will send the account information to the Payment Page once the user is ready to purchase a ticket.

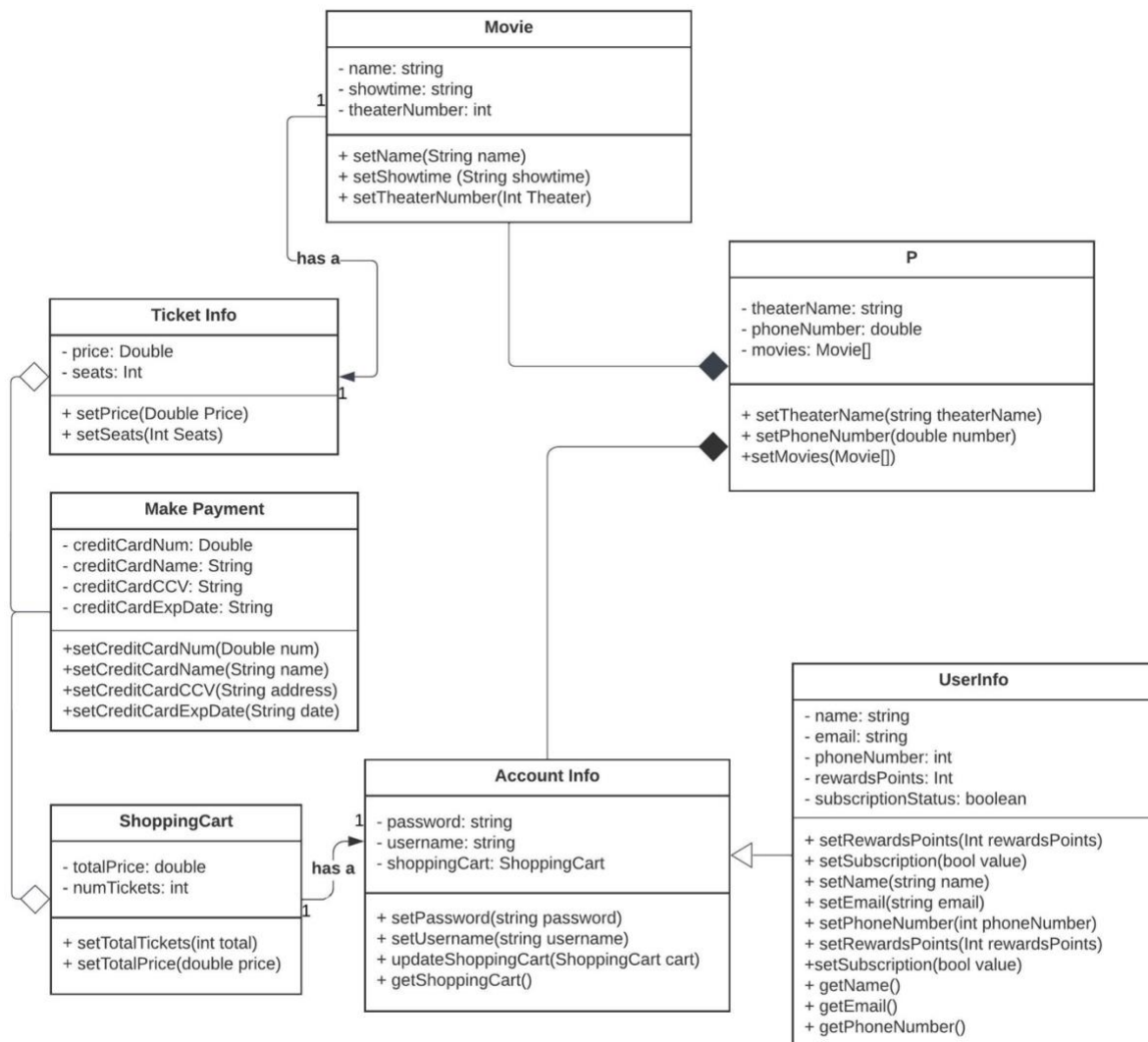
Move Schedule Page: Will be the interface for browsing a film to purchase at a particular cinema. It will take in the movie information from the database and display neatly all the films with their corresponding showtimes, prices, and locations.

Account Management Page: Will be the interface for modifying one's user account. It will take in the appropriate account information and will be modified according to the user's desires.

Movie Selection Page: Will be the interface for selecting a particular film to purchase. It will take in the particular movie information from the database and display neatly all the films with their corresponding showtimes, prices, and locations. It will then send the same movie information to the Seat Selection Page for further selection.

Seat Selection Page: Will be the interface for selecting a particular seat(s) for a particular film already chosen. It will take in movie information from the Movie Selection Page and set further variables such as tickets and seat. Then it will send out the movie information to the Payment Page for the finalization of the purchasing process.

The UML Class Diagram for the system will be the following:



The following are the more detailed descriptions of the classes, attributes, and operations of the system:

Movies [class]: This class defines the attributes of the films that will be screening in this cinema chain. They will have all the attributes that any film should have: a name, showtime, and screen number.

setName(string name): This operation will set the name by passing in a String as an argument.

setTime(string time): This operation will set the showtime by passing in a String as an argument.

setLocation(string location): This operation will set the cinema location by passing in a String as an argument.

setTheaterNumber(int number): This operation will set the screen number by passing in an integer as an argument.

Ticket Info [class]: This class defines the attributes of the tickets for a particular film at a cinema. They will hold a particular price as well as how many tickets are left for sale.

setPrice(Double price): This operation will set the price of a ticket for a particular film by passing a double as an argument

setSeats(int seats): This operation will set the number of seats left for sale for a particular film by passing an int as an argument.

Make Payment [class]: This class defines the attributes of the films that will be screening in this cinema chain. They will have all the attributes that any payment should have: credit card number, name of the credit card holder, address of the credit card holder.

setCreditCardNum(double num): This operation will set the credit card number of the holder when making a payment by passing in a double as an argument

setCreditCardName(string name): This operation will set the name of the credit card holder when making a payment by passing in a string as an argument

setCreditCardCCV(int num): This operation will set the credit card CCV number (the 3 digits on the back of a credit card) when making a payment by passing in an integer as an argument

setCreditCardExpDate(String date): This operation will set the credit card expiration date when making a payment by passing in a string as an argument

Account Info [class]: This class defines the attributes any general account for the cinema chain will have. They will have the attributes that any generic account, regardless of the user, should have including: a password, username, name, email, phone number, and an online shopping cart for the website.

setPassword(string password): This operation will set the password to a particular account made by passing in a string as an argument

setUsername(string username); This operation will set the username to a particular account made by passing in a string as an argument

updateShoppingCart(ShoppingCart cart): This operation will update the shopping cart of a particular user by passing in an object of the class type ShoppingCart (explained later) as an argument

getShoppingCart(): This operation will retrieve the online shopping cart of a particular account

User Info [class]: This class defines the attributes that an account should have in reference to a particular user. They will have attributes such as: name, email, and a phone number for the website.

setRewardsPoints(int rewardsPoints): This operation will set the num of rewards points a particular user has accumulated thus far by passing in an integer as an argument

setSubscription(bool value): This operation will set whether a particular user has the AMC subscription for the cinema chain by passing in a boolean as an argument

setName(string name): This operation will set the name of a particular user by passing in a string as an argument

setEmail(string email): This operation will set the email of a particular user by passing in a string as an argument

setPhoneNumber(double phoneNumber): This operation will set the phone number of a particular user by passing in a double as an argument

getName(): This operation will retrieve the name of a particular user

getEmail(): This operation will retrieve the email of a particular user

getPhoneNumber(): This operation will retrieve the phone number of a particular user

Shopping Cart [class]: This class defines the attributes that the online shopping cart would have of a particular user for a cinema chain. They will include attributes such as: the total price of the tickets, and the number of tickets purchased.

setTotalTickets(int total): This operation sets the current total number of tickets that the shopping cart for a particular user holds by passing in an integer as an argument

setTotalPrice(double price): This operation sets the current total price of the tickets that the shopping cart for a particular user holds by passing in a double as an argument

Movie Theater [class]: This class defines the attributes that a particular cinema in the cinema chain should have. They will include attributes such as: the name of the particular theater, the phone number to the particular theater, and the movies screening at the particular theater.

setTheaterName(string name): This operation sets the name of a particular theater in the cinema chain by passing in a string as an argument

setPhoneNumber(double number): This operation sets the phone number of a particular theater in the cinema chain by passing in a double as an argument

setMovies(Movies[]): This operation sets the list of films being screened at a particular theater by passing in an array of objects of the class type Movies (explained previously) as an argument

Development Plan and Timeline

The partitioning of tasks will be as such:

Luis Suarez – Programmer and Tester:

Will be responsible for heading the programming of the project as well of the testing and debugging.

Haliun Xu – Programmer and Quality Assurance:

Will be responsible for working alongside the head programmer with the programming of the project. Xu will as well be in charge for the quality assurance of all prototypes and finished products.