MVC-based Movie Theatre Seat Reservation Application

Programmer's Manual

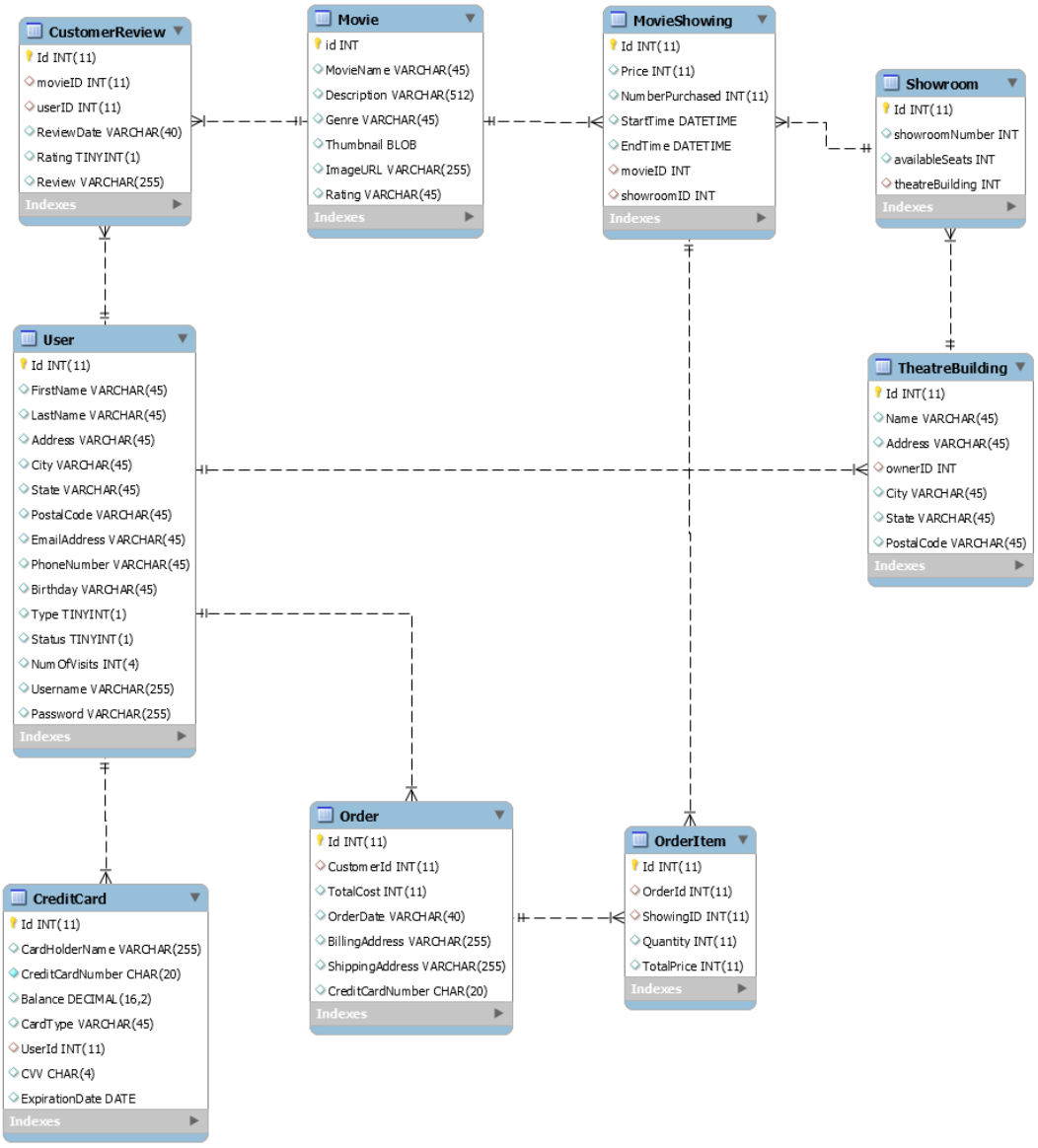
Table of Contents

Folders/Files/Funtions

OnlineTheatreManagementAndTicketPurchasingWebApp

Folder: Database

Database Schema Table:



DatabaseConnection.java

For connection to the database

DataCred.java

Stores the database credentials

CreditCardsDB.java

Stores the methods for accessing the database related to credit cards

String getCreditCardNumByOrder(int orderId, int customerId)

void updateCreditCard(Transactions aTransaction, java.math.BigDecimal totalCost)

Update the credit card of the user when the user makes a purchase or if there is a refund.

int cardNumberAndBalanceValidation(Transactions aTransaction, java.math.BigDecimal totalCost)

Takes the user credit card informations and returns the error number after validation checking. 0 for success, 1 for incorrect information entered, 2 for insufficient balance.

MoviesDB.java

Stores the methods for accessing the database related to the movies.

Movies getMoviesByMovieId(int movieId)

String getMovieRatingByName(String movieName)

String getMovieDescriptionByName(String movieName)

int getMovieIdByName(String movieName)

MovieShowingDB.java

ArrayList<MovieShowing> searchQuery(String searchMovieName, String searchMovieDate, String searchMovieTheatre)

Takes user's search query as input in the Home Page and produces a list of match movie information.

int getShowroomId(int showroomNumber, int theatreBuildingNumber)

Showroom getShowroomByShowroomId(int showroomId)

int getMovieShowingId(MovieShowing aMovieShowing, int movieId, int showroomId)

MovieShowing getMovieShowingByMovieShowingId(int movieShowingId)

int getNumberPurchased(int movieId, int showroomId, String startTime)

OrdersDB.java

Orders getOrderByOrderId(int orderId)

void addOrderItem(int orderId, int showingId, int quantity, int totalPrice)

int getOrderId(Orders aOrder)

addOrder(Orders aOrder)

ArrayList<Orders> retrieveOrder(int customerId)

Retrieve all orders of the user

ArrayList<OrderItems> retrieveOrderItem(int orderId)

Retrieve all order items of a specific order

void deleteOrderItem(int orderId, int movieShowingId, String startTime)

Delete a specific order item

ReviewsDB.java

ArrayList<Review> fetchReview(String movieName)

Get all the review of the movie at the movie details page

void commitReview(Review aReview, int movieId, int userId)

Add review of the movie.

TheatresDB.java

ArrayList<String> fetchAllTheatreName()

Get all of the theatres available in the database to display in the customer home page

int getTheatreId(String theatreName)

Theatres getTheatreByShowroomId(int showroomTheatreBuildingId)

UsersDB.java

Users getUsersByUserId(int userId)

int getUserId(String username)

void updatePassword(int userId, String newPassword)

Update password based on userId

void registerUser(Users aUser)

Add the user's information to the database

boolean isOwner(String Username)

Verify if the username belongs to the theatre owners or regular customers by checking the user's type in the database

boolean userExist(String username)

Verify if the username exist in the database

boolean passwordValidation(Users aUser)

Verify if the password of the given username is correct

int reauthenticateUser(Users aOriginalUser, Users aUser)

Reauthenticate the user in the session before sending sensitive data

void validateUser(Users aUser, HttpServletRequest request, HttpServletResponse response)

Validates user and if any error occurs, redirect user to Registration page, else go to the customer home page.

Folder: Model

EmailUtility.java

Utility class to send email via SMTP. (Implemented this for bonus)

Movies.java

Manage information of Movies such as movie title, description and MPAA rating

Theatres.java

Stores theatre information such as its name and address

Showroom.java

Stores the capacity of the room and a reference to the theatre where it is located

MovieShowing.java

Stores a reference to the movie object, a relevant showroom object, number of purchased seats for the showing and the cost of the showing

Review.java

Stores the content, rating of a review, a reference to the user submitting the review , reference to the movie object.

Validates the contents of the review do not exceed maximum size (255 characters).

Transactions.java

Stores the balance and validates credit card account details.

Users.java

Stores user information such as name, address, phone number, etc.

String hashPassword(String password)

Hash a clear-text password using specified algorithm in Java security API using SHA-256

String generateSalt()

Generate random string to append to the password

String hashAndSaltPassword(String password, String saltValue)

Method for salted hashing a password

ShoppingCart.java

Stores showtime information to display in the shopping cart.

Orders.java

Stores order information.

OrderItems.java

Stores order items information.

Folder: Servlets (Controller)

Login.java

Handling requests from the login page. Username and password from Login.jsp will be send to UsersDB's validateUser method for processing. If successful, it will forward user to CustomerHomePage.jsp. If not, it will redirect user to registration page. Synchronized techniques are implemented here to count the number of user's access.

Register.java

Handling requests from the register page. Username and password from Registration.jsp will be send to addUser method in UserDB class. Email will be sent to the user's email. It will forward back to login page after registering.

TheatreAndMovieSearchQuery.java

Retrieve the search query passed by the Customer Home Page.

Retrieves the matching movie and showtimes using the MovieShowingDB's searchQuery method.

Redirects user to the Movie Search Result jsp page with the search results.

MovieSearchResults.java

Receives the movie data and complete information from the Movie Search Results page.

Retrieve the reviews from the database to display in the Customer Review section in the View Movie Details and Selection page.

Redirects user to the View Movies Details and Selection page with the movie details.

CustomerReview.java

Receives the review and movie models from the Customer Review page and adds the review to the database using ReviewDB's commitReview method.

On success, send back the html data along with the table tags to dynamically view the latest review submitted by the customer through AJAX call.

On failure, send back error message.

UpdateShoppingCart.java

Creates a shopping cart session object to be used to store the shopping cart item temporarily.

Display meaningful message on successful or not successful update of the cart.

Handles remove item from the shopping cart (removing from the shopping cart session object)

Computes the total cost of the entire shopping cart

Redirect user to View&CheckoutShoppingCart to see the updated shopping cart when the user clicks on the check out button.

PlaceOrder.java

Reviews order details from ConfirmOrder.jsp and places order to the databse.

CustomerTransactionConfirmation.java

Receives the credit card details from Customer Transaction page.

Verifies the credit card details using CreditCardDB method

On success, order is placed and the order is added to the database with the customer's credit card is charged. Shopping cart session cleared to empty after that.

Redirects the customer to Customer Transaction Confirmation page with order details with the Order number shown.

ViewOrders.java

Find all orders for the customer using session information of the customer.

Fetches all customer's orders using OrdersDB method.

Redirects user to the View Orders page

ManageOrder.java

Fetches detailed information for the Order received from the ViewOrders page.

Forwards information to Manage Order page.

ManageOrder.java

Fetches detailed information for the Order received from the ViewOrders page.

Forwards information to Manage Order page.

CancelOrder.java

Receives the orderItem to be cancelled from Manage Order page.

Pass information and direct customer to Cancel Order page.

CancelOrderTransaction.java

Delete order item using deleteOrderItem method in OrdersDB class.

Equivalent amount of the cost of the cancelled order item is refunded to the credit card account that was used to purchase.

Redirects the customer to the Cancellation Confirmation page.

AddMovie.java

Receives the movie information from the Add Movie jsp page and create a new movie. Then store it to the Movies DB class.

MovieSearchResult.java

Receives the movie search information from the owner home page and retrieves matching movies from the MovieDB class. After that pass the list of the matching movie to the MovieSearchResults JSP page.

MovieDetails.java

Get the movie from the Movie Search Result JSP. Then get the movie by using the method in MovieDB class and pass the object movie to the movie details JSP page.

MovieDetailsUpdate.java

Get the movie object movie from the movie detail page. Owner can change or update the movie. Owner will redirect back to the Owner Home page.

ViewTheatreDetails.java

Uses session information to find all the Theatres owned by the use. Then past the list of the theatres that the Owner own to the View Theatre Details jsp.

ViewShowrooms.java

Receives the theatre number from the View Theatre Details jsp page and fetches the theatre information from the TheatreBuildingDB. Then pass all the information to the View Showrooms JSP

MangeShowtimes.java

Receives the theatre and showroom number from the view showrooms jsp page. After that, Retrieves information about all movies playing in this showroom from the database (from the current date/time forward) using the MovieShowingsDB class. Then send all the information needed to the Manage Showtimes jsp

AddShowtime.java

Receives the showtime information from the AddShowtime JSP page. Adds a showtime in the given showroom for the movie ID at the given time and date. Send the message Success or failure to the Add Showtime Confirmation page.

CancelShowing.java

Receives a showtime ID from the manage showtimes jsp page. Removes the given showtime from the database and updates all tables. Refunds credit cards for each purchased ticket. Send the information about the show times that has been delete.

CancelPassing.java

Get all the information need for the cancel from manage showtimes page. Send all information need to the Cancel Showing jsp.

ChangePassword.java

Changed the password of the user in the database to a new password if the current password matched.

Folder: WebContent (View)

Login.jsp

User login and log

> validateForm() method is invoked to check for empty strings when user submit username and password.

Registration.jsp

New user registration

CustomerHomePage.jsp

Customer home page to enable searching for movies based on the name of the movie

> validateForm() method is invoked to check for empty strings when user submit movie search and movie date. It also checks for date format.

MovieSearchResults.jsp

To view the search results from the customer home page.

MovieDetailsSelection.jsp

To view movie details (showtimes and theatre locations, prices, and seat location) along with the customers' review on the movie. AJAX is implemented to call the UpdateShoppingCart servlet.

On success, the movie with the quantity will be stored in the shopping cart.

On failure, an error message will appear.

It also allows users to input their comments about the movie. AJAX is implemented to call the CustomerReview servlet.

On success, the page will be updated with the latest review submitted by the user.

On failure, an error message will appear.

getData() method is invoked to call the UpdateShoppingCart servlet.

addReview() method is invoked to call the CustomerReview servlet.

ViewAndCheckoutShoppingCart.jsp

Viewing the shopping cart.

CustomerReview.jsp

Submit review of a movie.

ConfirmOrder.jsp

Purchase order transaction.

AJAX is implemented to call the bank servlet with sending data to each other over a network using HTTP.

validateForm() is invoked to validate ticket quantity.

confirm_function() is invoked to send request to the bank application to validate the transaction details.

It will show success/failure message. place_order_function() will be invoked.

place_order_function() is invoked to call Place Order servlet to save the order.

ViewOrders.jsp

To view all currently purchased tickets

ManageOrder.jsp

To manage orders

CancelOrder.jsp

To cancel orders

CancellationConfirmation.jsp

To confirm cancellation of an order

AddMovie.jsp

To add the movie based on owner choice

MovieSeaerchResutsOwner.jsp

Show the list of the movie exist on the data base

MovieDetails.jsp

View the full information of a movie

ViewTheatreDetails.jsp

View the full information of all the theatre own by owners

ViewShowrooms.jsp

Display the showroom details in a suitable format

ManageShowtimes.jsp

Display all the showtimes of all the movie playing In that theatre in suitable format

AddShowtime.jsp

Add more show time to the theater

AddShowtimeConfirmation.jsp

Confirm the show time has been add to the Theatre. Print failure message if the time is conflict

CancelShowing.jsp

Cancel the show time of a movie in that Theatre.

CancellationConfirmationShowtime.jsp

Show the message that the show time has been cancel successfully

AccountSettings.jsp

Allows user to change the current password.

Folder: WEB-INF

web.xml

Login.jsp will be called when the project ran on server

Page will be redirected to error.jsp when error code 404 occur

Page will be redirect to exception.jsp when exception type java.lang.Throwable occur

context-param (email of the owner of the web app) is used on each page to assist users with enquiries.

SMTP settings for sending notification emails after registration is stored here

SSL settings can be enabled/disabled here.

Folder: images

Contains the movie poster/thumbnail

BankWebApp

Folder: Database

DatabaseConnection.java

For connection to the database

DataCred.java

Stores the database credentials

BankDB.java

Stores the methods for accessing the bank database related to credit cards

void updateCreditCard(Bank aBank, java.math.BigDecimal totalCost)

Update the credit card of the user when the user makes a purchase or if there is a refund.

int cardNumberAndBalanceValidation(Bank aBank, java.math.BigDecimal totalCost)

Takes the user credit card informations and returns the error number after validation checking. 0 for success, 1 for incorrect information entered, 2 for insufficient balance.

Folder: Model

BankModel.java

Stores the balance and validates credit card account details.

Folder: Servlet (Controller)

Bank.java

Receives the credit card details from Customer Transaction page.

Verifies the credit card details using CreditCardDB method

On success, order is placed and the order is added to the database with the customer's credit card is charged. Shopping cart session cleared to empty after that.

Updates page with order details with the Order number shown.