

Cinema Management System

YANG Jiahao

November 17, 2024

1 Overall Project Description

The project is a simple simulation of a cinema management system with two primary roles: Administrator and User. I have implemented several functionalities for these roles. The main design goal is to implement basic features from a practical perspective, such as movie release, movie removal, user ticket purchasing, and user reviews.

2 Requirements Analysis

The system has two primary roles: Administrator and User, each with specific functionalities.

2.1 Administrator Functionalities

- **View Movie Information:** The administrator can view details of movies currently showing in the cinema.
- **Release a Movie:** The administrator can add new movies to the cinema's schedule.
- **Remove a Movie:** The administrator can remove movies from the cinema's schedule.
- **Update Movie Screening Information:** The administrator can modify the screening times and other details of movies.
- **Display User Reviews:** The administrator can view user reviews for movies.
- **Synchronize User Ratings:** The administrator can update the average rating of movies based on user reviews.

2.2 User Functionalities

- **Get Movie Information:** Users can view details of movies currently showing in the cinema.
- **Rate a Movie:** Users can rate and review movies they have watched.
- **Book Movie Tickets:** Users can purchase tickets for movies.

- **View Own Review History:** Users can view their own review history.
- **View Own Booking History:** Users can view their own booking history.
- **View Movie Reviews:** Users can view reviews of a specific movie by other users.

3 Database and SQL Design

Navicat can be used to directly observe the structure of the database. Figures 1 to 6 are the names and attribute information of each table in my database


字段	索引	外键	检查	触发器	选项	注释	SQL 预览							
名					类型		长度	小数点	不是 null	虚拟	键		注释	
► BookingID					int				<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1			
UserID					int				<input type="checkbox"/>	<input type="checkbox"/>				
ShowtimeID					int				<input type="checkbox"/>	<input type="checkbox"/>				
BookingDate					datetime				<input checked="" type="checkbox"/>	<input type="checkbox"/>				

Figure 1: Main attributes of the bookings table.

字段	索引	外键	检查	触发器	选项	注释	SQL 预览				
名					类型	长度	小数点	不是 null	虚拟	键	注释
► ReviewID					int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	
UserID					int			<input type="checkbox"/>	<input type="checkbox"/>		
MovieID					int			<input type="checkbox"/>	<input type="checkbox"/>		
Rating					int			<input type="checkbox"/>	<input type="checkbox"/>		
Comment					text			<input type="checkbox"/>	<input type="checkbox"/>		
ReviewDate					datetime			<input checked="" type="checkbox"/>	<input type="checkbox"/>		

Figure 2: Main attributes of the reviews table.

字段	索引	外键	检查	触发器	选项	注释	SQL 预览																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
----	----	----	----	-----	----	----	--------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Figure 3: Main attributes of the users table.


字段	索引	外键	检查	触发器	选项	注释	SQL 预览				
名					类型	长度	小数点	不是 null	虚拟	键	注释
► AdminID					int			<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	
Username					varchar	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Password					varchar	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>		

Figure 4: Main attributes of the admins table.

字段	索引	外键	检查	触发器	选项	注释	SQL 预览
名							
ShowtimeID					int	长度	不是 null 虚拟 键 注释
MovieID					int	长度	
StartTime					datetime	长度	
EndTime					datetime	长度	

Figure 5: Main attributes of the showtimes table.

字段	索引	外键	检查	触发器	选项	注释	SQL 预览
名							
MovieID					int	长度	不是 null 虚拟 键 注释
Title					varchar	255	
Director					varchar	255	
Actors					text		
Genre					varchar	255	
ReleaseDate					date		
IsListed					tinyint	1	
AverageRating					decimal	3 2	

Figure 6: Main attributes of the movies table.

4 Front and Back-End Design and Implementation

Figures 7 to 9 show the implementation of the main functions of the program. The code marked in red is the front-end code, and the code in black but not marked in red is the back-end code.

Requirements	The function that implements it	Function Introduction
Admin Registration	<code>admin_register()</code>	Accept information used by users for registration
	<code>unique_admin_id_detection()</code>	Check if duplicate IDs occur
	<code>admin_register()</code>	Submit a request to create new data to the database
User Registration	Similar to "Admin Registration"	Similar to "Admin Registration"
Admin Log in	<code>admin_log_in()</code>	Accept user's login information
	<code>admin_log_in()</code>	Query the database to see if there is a matching user
User Log in	Similar to "Admin Log in"	Similar to "Admin Log in"

Figure 7: Login and Registration implementation.

Requirements	The function that implements it	Function Introduction
User review movie	review_movie()	Get content information of user review
	movie_from_name_to_id()	Get movie_id to help query
	unique_movie_id_detection()	Check whether the relevant ID exists
	unique_review_id_detection()	Check whether the relevant ID exists
	review_movie()	Insert data into the table based on user-submitted information
User view reviewed history	view_my_review_hisory()	Collect confirmation information requested by the user
	view_my_review_hisory()	Retrieve data for users from the reviews table
User book tickets	book_tickets()	Get content information of user want to book
	movie_from_name_to_id()	Get movie_id to help query
	unique_movie_id_detection()	Check whether the relevant ID exists
	unique_book_id_detection()	Check whether the relevant ID exists
	review_movie()	Retrieve the relevant information of the showtime table and movie table, and add data to the bookings table after the user selects it.
Adimin / User view all movie informations	Already shown above	Already shown above
User view booking history	Similar to "User view reviewed history"	Similar to "User view reviewed history"
User view all review	Similar to "Admin view all review"	Similar to "Admin view all review"

Figure 8: User functionalities implementation.

Requirements	The function that implements it	Function Introduction
Admin update movie score	update_score()	Accept user's confirmed update request
	movie_from_name_to_id()	Get movie_id to help query
	unique_movie_id_detection()	Check whether the relevant ID exists
	update_score()	Update scores and sync to database
Admin view all review	view_all_reviews()	Accept user query request and get the requested movie name
	movie_from_name_to_id()	Get movie_id to help query
	unique_movie_id_detection()	Check whether the relevant ID exists
Admin modify showtime	view_all_reviews()	Get the corresponding review information and print it
	modify_showtime_info()	Get the information the user wants to modify, movie name, and new showtime
	movie_from_name_to_id()	Get movie_id to help query
	unique_movie_id_detection()	Check whether the relevant ID exists
Admin remove movie	modify_showtime_info()	Submit changes and update database results
	remove_movie()	Accept movie titles that need to be removed
	movie_from_name_to_id()	Get movie_id to help query
	unique_movie_id_detection()	Check whether the relevant ID exists
Admin add movie	unique_showtime_id_detection()	Check whether the relevant ID exists
	remove_movie()	Modify information in the movie table and delete records in the showtime table
	Similar to "Admin Log in" & "User Log in"	Similar to "Admin Log in" & "User Log in"
	Similar to "Admin Log in" & "User Log in"	Similar to "Admin Log in" & "User Log in"
Adimin / User view all movie informations	get_movie_info()	Get the user's view request type
	get_movie_info_all()	Find and return all movies in the database to help the user return to the initial page
	get_movie_info_off()	Find and return offline movies in the database to help the user return to the initial page
	get_movie_info_show()	Find and return listed movies in the database to help the user return to the initial page

Figure 9: Administrator functionalities implementation.

5 How to Use the System

Refer to the demo video for the most accurate instructions, as it demonstrates all the methods in detail.

Note that the code connects to a local database. If you are running the system on a different machine, you need to modify the `connect_to_db()` method at the beginning of the code. Replace the database connection information with the appropriate details for your database and ensure that the required tables are created in your database.

If you have successfully deployed the database, you can now follow the program's instructions to use the system.