SmartCine: Movie Theater Reservation System

Xiaoyang Fei[1], Xinyuan Zhao[2]

[1,2]Northeastern University

fei.xiao@northeastern.edu, zhao.xinyuan@northeastern.edu

Abstract

This is a final project report for Northeastern University 2023 Fall semester DS 5110. In order to improve user experience when going to the movies and to make theater operators' jobs easier, this study describes the creation of an online application for an easy-to-use movie theater reservation and admin management system. React and Material-UI (MUI and MUX) are utilized in the front-end design for building a user-friendly interface. With Python Flask, the back-end has been created with Axios for HTTP requests and Router-DOM for navigation. MySQL databases are used for organizing data. The report covers performance analysis, input from users, and a summary of the development process, issues and remedies. It closes with suggestions regarding additional enhancements that focus on improving the functionality and value of the application.

Introduction

SmartCine is a movie theater reservation and management system developed by Xiaoyang Fei and Xinyuan Zhao. It was initially developed with the goal to make a better movie-going experience for theater users and also help the administrative processes for theater administrators. This system is developed using modern reliable software development tools and technologies, such as MySQL for database management, Python Flask for back-end software development and Rect Javascript, Material-UI (MUI and MUX) for front-end user interface. SmartCine is a user-centric design. The system offers many features that can actually help theater customers to enjoy their movies:

- Search Functionality: Users can search for all movies that are contained in the database, this system will provide them with the access to all movie detailed information such as movie title, movie duration, movie director and movie release date etc. They can have a wide range of options.
- Detailed Show Information: Each movie listing is accompanied by detailed information, including showtimes, cast details and more, enabling users to make informed decisions about which show they are able to reserve.
- Ticket Reservation: The reservation feature is streamlined to ensure a hassle-free process, from selecting seats to finalize the booking.
- Registration and Payment Integration: Users can register their customer accounts with their email. After successful registration, they will automatically get a card within the system for earn points, enhancing the convenience of the ticket redeem process.
- Electronic Tickets: In keeping with modern digital trends, SmartCine issues electronic tickets, facilitating a paperless and eco-friendly approach.

SmartCine is equally equipped with features tailored for theater administrators, enabling efficient management and insightful analytics:

- Movie Data and Schedule Management: Administrators can easily manage movie data and schedules, ensuring that the most current information is always available to users.
- User Data Management: Administrators can not register their own accounts as a system safe approach. Instead, when the system owner created an administrative account, administrators can use that account to manage all user data such as customers data.
- Insightful Analytics: The system provides valuable insights, such as identifying popular movies and generating monthly revenue statistics, aiding in strategic decision-making.

SmartCine's major objective is to completely transform the traditional movie theater reservation experience by integrating technical advances with user-friendly interfaces. The seminar will go through the SmartCine development process, architectural design, and functioning, emphasizing how it fulfills the needs of both users and administrators. The paper's scope includes the idea of the system, technological choices, implementation strategies, and the predicted contribution to the movie-going experience and theater management. SmartCine's breakthrough approach to the industry of entertainment is projected to set new benchmarks by providing a seamless and effective reservation system. SmartCine is positioned to alter how moviegoers and theater owners participate with and run business by addressing current gaps in movie theater reservations systems and creating a platform that is both user-friendly and administratively powerful.

Database design

The SmartCine database design, as elucidated in the Jupyter Notebook, represents a sophisticated approach to structuring and managing a cinema-centric database system.

ERD

The ERD is a detailed schematic representation, encompassing a broad range of entities essential for cinema operations. It includes:

Movies: With attributes like title, genre, director, and duration, encapsulating the essence of each film.

Screenings: Linking movies to specific showtimes and halls, highlighting logistical details.

Customers: Capturing customer demographics and preferences for personalized services.

Tickets: Detailing ticket transactions, seating, and pricing.

Reviews: Providing a platform for customer feedback on movies.

The relationships between these entities, as shown in the ERD, are integral to the database's functionality, ensuring a seamless flow of information across different modules.

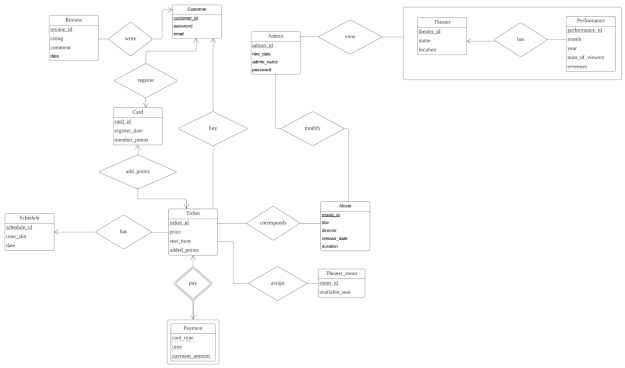

Figure 1: Final ERD of SmartCine Database
(This illustrates the relationships between Movies, Screenings, Customers, Tickets, and Reviews.)

Design Decisions

The Jupyter Notebook reflects several critical design decisions:

Relational Structure: The choice of a relational structure allows for complex queries and transactions, which are pivotal in a dynamic cinema environment.

Data Integrity and Reliability: Implementing primary and foreign keys and other constraints ensures consistency and accuracy of data.

Scalability and Adaptability: The design is scalable, considering potential future expansions, such as incorporating new movie genres or additional screening formats.

Normalization Procedures

The normalization process, deeply rooted in the ERD, was methodically executed:

First Normal Form (1NF): Tables were structured to eliminate repeating groups, ensuring atomicity in data entries. This is evident in the SQL scripts for table creation.

Second Normal Form (2NF): The database was restructured to achieve full functional dependency on primary keys, thereby eliminating any partial dependency issues.

Third Normal Form (3NF): The progression to 3NF involved removing transitive dependencies, ensuring that all data is directly dependent on primary keys, enhancing data integrity and query performance.

The Jupyter Notebook details the implementation of these design principles through SQL scripts and Python codes, highlighting the practical application of theoretical database design concepts in real-world scenarios.

Data collection

We utilized the "Netflix Shows" dataset (which were created by Shivam Bansal accessible at https://www.kaggle.com/datasets/shivamb/netflix-shows/data) as the main source of data for our web application. The dataset has an extensive collection of data on a wide range of Netflix programs, including title, director, release date, year of release, ratings, duration, and genres. To get around the dataset's size and scope limitations, we used a two-pronged approach:

1. Data Expansion: To keep the dataset rising we added new data sources. This included locating and merging complementary datasets that gave more information on the Netflix series, such as movie_id and others. To guarantee consistency across the merged dataset, careful data rearrangement and numerical standardization were required during the integration process.

2. Data Cleaning and Transformation: The raw data has been thoroughly cleaned and converted. This engaged standardizing date and duration formats, movie_id, deduplicating and missing records, and addressing naming convention oddities.

3. Data Enrichment: We updated the data by introducing calculated fields where they were necessary. To simplify more sophisticated inquiries, we included combined information.

4. Normalization: We normalized the data to allow effective data retrieval and storage. This involved dividing the dataset into multiple parts.

5. Final Data Integration: The final step involved the integration of the augmented and modified dataset into our MySQL database. This process was executed with a focus

on maintaining data integrity and ensuring that the expanded dataset was seamlessly incorporated into our existing database structure, allowing for more robust data analysis and a richer user experience on our web application.

Application description

The SmartCine application is an advanced platform for cinema management, developed using Python for the user interface and SQL for backend data handling.

• User Interface (UI): The Python-based UI is intuitively designed, allowing users to seamlessly navigate through various functionalities. It facilitates easy management of cinema operations like adding new movies to the database, scheduling movie screenings, overseeing ticket bookings, and collating customer reviews.

• Interactive Data Management: Users can actively interact with the database by running custom SQL queries. This feature enables them to retrieve specific data, perform in-depth analysis, and generate custom reports. Additionally, the interface supports direct data manipulation, allowing users to insert, update, and delete records within the database tables. This ensures that the database remains up-to-date and reflective of the current state of cinema operations.

• Backend Functionalities:

- SQL Functions: Functions such as "count_customer" are used to streamline specific data operations, enhancing efficiency in data handling and retrieval.
- Views: The application includes views like "ViewMovie", "ViewCustomer", and "ViewTicket", which are designed to provide organized access to specific data segments. These views are crucial for quick and efficient data retrieval, minimizing the complexity of direct table queries.
- Stored Procedures: Stored procedures including "AddAdmin", "UpdateAdminPassword", and "DeleteAdmin" offer a secure and efficient way of executing complex database operations. These procedures encapsulate business logic in the database layer, facilitating maintenance and scalability.
- Triggers: Triggers, such as "validate_rating_before_insert", are implemented for automatic data validation. They ensure data integrity by enforcing business rules at the database level, such as validating input data before it is inserted or updated in the database.

• Data Visualization and Analysis: The application integrates Python's data visualization libraries, providing powerful tools for visual data analysis. These tools are instrumental in uncovering trends and patterns in movie popularity and customer preferences, offering valuable insights for strategic decision-making in cinema management.

The SmartCine application is a testament to the effective integration of front-end usability and back-end sophistication, providing a comprehensive solution for modern cinema management.

Conclusions

The SmartCine project effectively demonstrates the synergy between robust database systems and intuitive user interfaces, highlighting the practicality and efficiency of SQL and Python in cinema operations. It brings to light the critical role of advanced database features such as custom functions, views, and triggers in enhancing overall functionality and user experience. Additionally, the utilization of data visualization tools underscores the significance of data analytics in understanding customer preferences and identifying trends within the cinema industry.

Future Directions

• Mobile Application Development: Expanding the system to include a mobile app interface could significantly increase accessibility and convenience for users.
• Integration of Machine Learning: Implementing machine learning for personalized recommendations and predictive analytics could revolutionize user experience and operational efficiency.
• Real-Time Data Processing: Incorporating real-time data streaming would provide dynamic updates, offering up-to-the-minute insights for both customers and management.
• Expansion of Features: Adding features like online reservations and loyalty programs could enhance customer engagement and retention.

Advice for Future Development

• Future iterations should focus on thorough testing and iterative design to refine both the backend and frontend aspects of the application.
• Keeping abreast of technological advancements and integrating emerging technologies will be crucial in maintaining the application's relevance and efficiency in the evolving cinema industry landscape.

These conclusions and future directions suggest a pathway for the SmartCine application's evolution, highlighting its current successes and potential for future growth and innovation in cinema management.

References

Website or online resource

Shivam B. (2021). Netflix Shows [Data set]. Kaggle. https://www.kaggle.com/datasets/shivamb/netflix-shows/data.