



The updated version of our SWA diagram contains two main changes. The first one was an addition of a Movie database that we realized we were missing and is essential for the ticketing system. The movie database is planned to contain a table including details of the name, genre, length, rating, and description of each movie offered. In addition to that we added an application server that is essential for handling customer requests and validating user input.

Our movie theater ticketing system is designed to manage and process persistent and sensitive data related to customer information, showtimes, and seat availability. To achieve this, we have chosen to use a SQL database to handle structured data due to its excellent support for structured data and data integrity. With this approach, we can store data securely and access it quickly and efficiently whenever needed. To ensure efficient data management, we have implemented an application server that acts as an intermediary between the user interface and the database. This design allows the application server to handle multiple user requests simultaneously while the database manages the persistent data. We have opted to use a single SQL database to manage all data related to the movie theater ticketing system. The data is split into three primary tables: movies, theaters, and showtimes, providing a logical separation of data. This structure optimizes data retrieval speed and ensures that queries are performed

efficiently. Furthermore, the architecture simplifies data management, ensuring that all data related to the ticketing system is stored in one place. In summary, our movie theater ticketing system uses a SQL database to manage structured data, an application server to handle user requests, and a single database to store all data related to the system. This approach provides high data integrity and reliability, scalability, and efficient data retrieval.