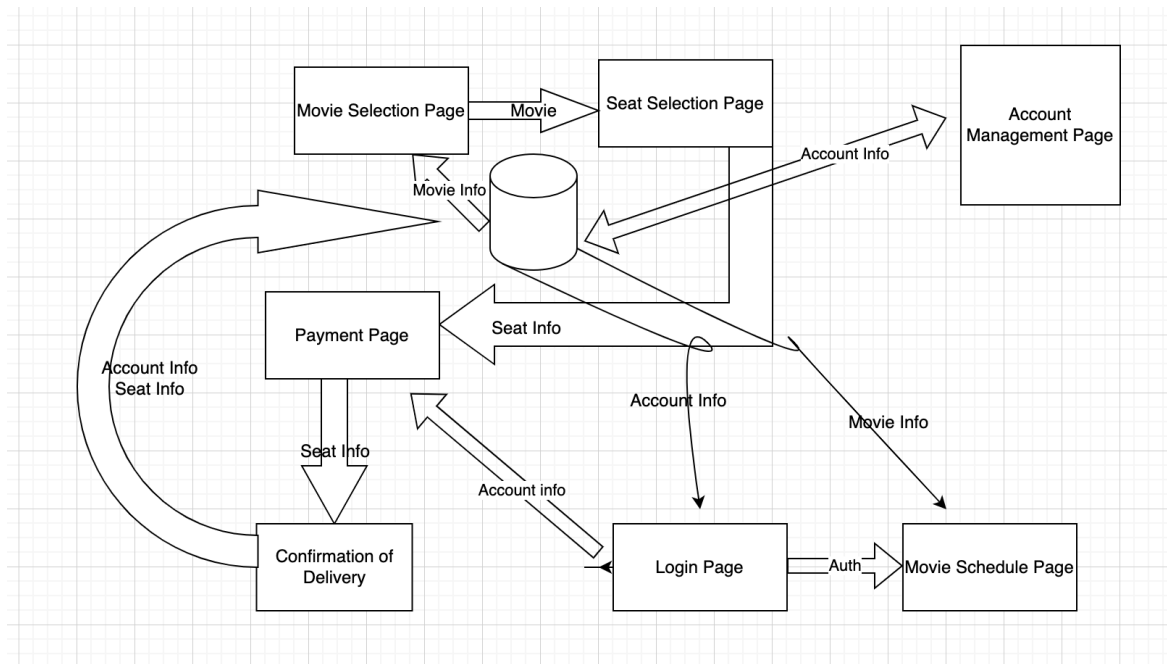


Architecture Design w/ Data Management

Prepared by: Grace Peebles, Victoria Nguyen, Gilbert Reyno, Cameron Croom, Mark Langell
April 15, 2023

Software Architecture Diagram:



In the Software architecture diagram, each connector has a label which points to the next component that determines what the system's next step is. For example, when on the Movie Selection Page, the user selects a movie and is then directed to the Seat Selection Page. We have designed the diagram to reflect what pages the user will be directed to after completing certain steps of the ticket purchasing process. In order for the user to have the best experience possible we felt that it would be easiest for them to select a movie then their seat then move on to the more specific details like picking a theatre and their payment method. There are many ways that the user could explore the website but this is the most straight forward and likely approach that the user would take.

Data Management Strategy:

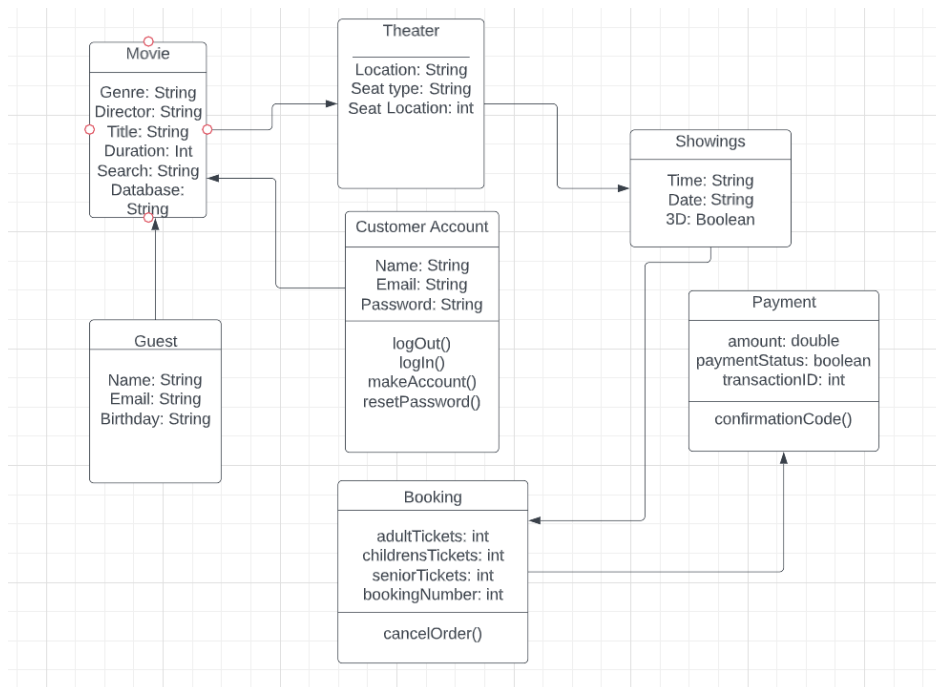
Customers input their personal information into our movie ticketing system and we need to be confident that we can keep their potentially sensitive data safe and organized. In order to maintain a secure system we will be implementing a NoSQL database. NoSQL databases allow

new data to be easily inserted and does not require any prior steps which is a huge benefit. As a movie ticketing system, there will constantly be movies that need to be added and removed from our database depending on what movies we are showing in theatres that month. The downside of NoSQL is it is not the best fit for huge transactional purposes however this is not a major issue for our system given that movie tickets are not typically bought in bulk and even when they are the purchases are not large enough to cause any issues.

Design Decisions:

We plan on having seven different databases. This will include a different database for the movie, theater, movie showings, customer account, guest, booking, and payment information. Each database will include a collection of data that corresponds to its purpose. The movie database will have information about the genre, movie director, title, and duration. The theater will withhold the information that includes its location and the seat location as well as the seat type. The customer account database takes care of the information of the customer such as their name and email in order to make an account. The showings database includes the time and date of the movie, as well as whether it is a 3D movie or not. The guest database exists for when a user does not want to create an account with the ticketing system, but it still holds their name and email. The booking database contains the integer amount of each type of child, regular, and senior ticket along with the booking number. The payment database enables us to make a payment for the tickets selected and provides a confirmation code once the payment is completed.

Database Design Diagram:



Trade Off Discussion:

Although NoSQL is still the best decision for our system SQL does have some features that we are going to be missing out on. For example, SQL has high transactional applications. However, adding new data requires some changes to be made like backfilling and alternative schemas which is a more complicated alternative to what the NoSQL database has to offer. Additionally, SQL databases are not suited for storing large data sets. There is no way to tell how many people will be using the movie ticketing system and therefore we cannot assume that SQL will be sufficient enough to handle all of the users' personal information. Another pro to using an SQL database is its aged technology. This could be beneficial because there are more people who know how to work with SQL databases and more research done on them. With that being said NoSQL is a relatively young technology and there are not as many people who are familiar with it but its ability to have dynamic schema for unstructured data and other improvements make up for it. There upsides and downsides to using both SQL and NoSQL databases but in this particular case the NoSQL database is the best choice.