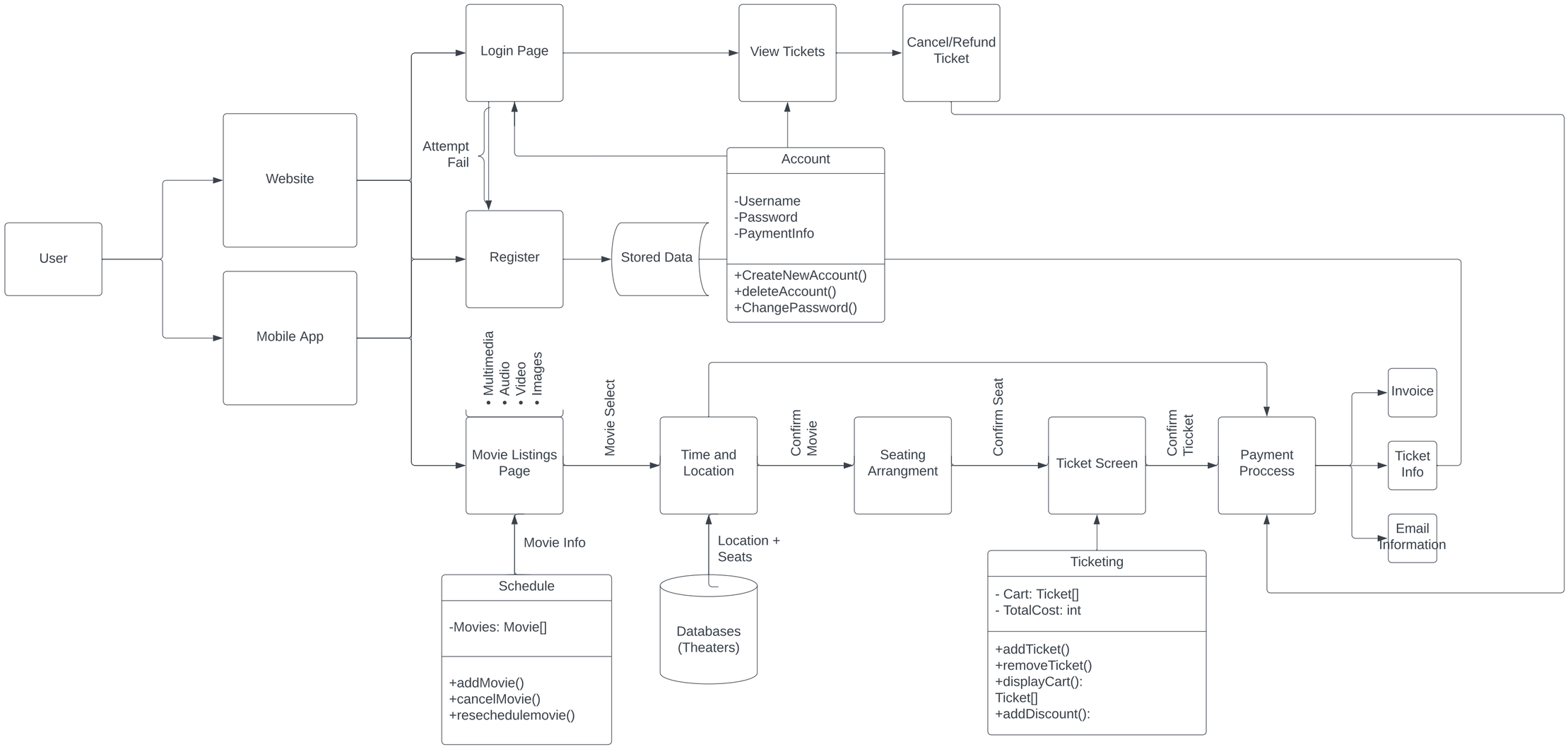
Login Page
View Tickets
Cancel/Refund Ticket
Website
Attempt Fail
Account
-Username
-Password
-PaymentInfo
+CreateNewAccount()
+deleteAccount()
+ChangePassword()
User
Register
Stored Data
Mobile App
• Multimedia
• Audio
• Video
• Images
Movie Select
Confirm Movie
Confirm Seat
Confirm Ticcket
Invoice
Movie Listings Page
Time and Location
Seating Arrangment
Ticket Screen
Payment Proccess
Ticket Info
Email Information
Movie Info
Location + Seats
Schedule
-Movies: Movie[]
+addMovie()
+cancelMovie()
+resechedulemovie()
Databases (Theaters)
Ticketing
- Cart: Ticket[]
- TotalCost: int
+addTicket()
+removeTicket()
+displayCart(): Ticket[]
+addDiscount():

Data Management Strategy:

Our data management strategy involves using a SQL database management system to store and manage our data. We chose this approach because SQL is a well-established and widely used technology that is reliable and secure. Additionally, it provides powerful tools for querying and analyzing data, making it easier for us to develop and maintain our system.

We have chosen to split our data across three databases for the following reasons:

Ticketing: This database contains information about the tickets purchased by users. It is separated from the Schedule database to allow for more efficient querying of data and to improve data security.
Schedule: This database contains information about the movies being shown at each theater. It is separated from the Ticketing database to allow for more efficient updating of data and to improve data security.
Account: This database contains information about user accounts and payment information. It is separated from the other two databases to improve data security and to allow for easier management of user accounts.
We have also designed our databases to be scalable, so that we can add more data as our system grows. This involves using normalized tables and setting up indexes to improve query performance.

Tradeoff Discussion:

Our choice to use SQL databases for our data management strategy comes with several tradeoffs. One of the main tradeoffs is the complexity of setting up and maintaining the databases. SQL databases require a significant amount of configuration and administration, which can be time-consuming and costly.

Another tradeoff is the potential for data loss in the event of a database failure. SQL databases store data on disk, which makes them vulnerable to disk failures and other types of hardware issues. This can result in data loss if proper backups are not in place.

Despite these tradeoffs, we believe that using SQL databases is the best choice for our system. SQL databases provide a high level of reliability and security, as well as powerful tools for managing and analyzing data. Additionally, there is a wealth of resources and support available for SQL databases, making them a widely accepted and well-understood technology.