Documentation of theater App

Architecture diagram for theater webapp

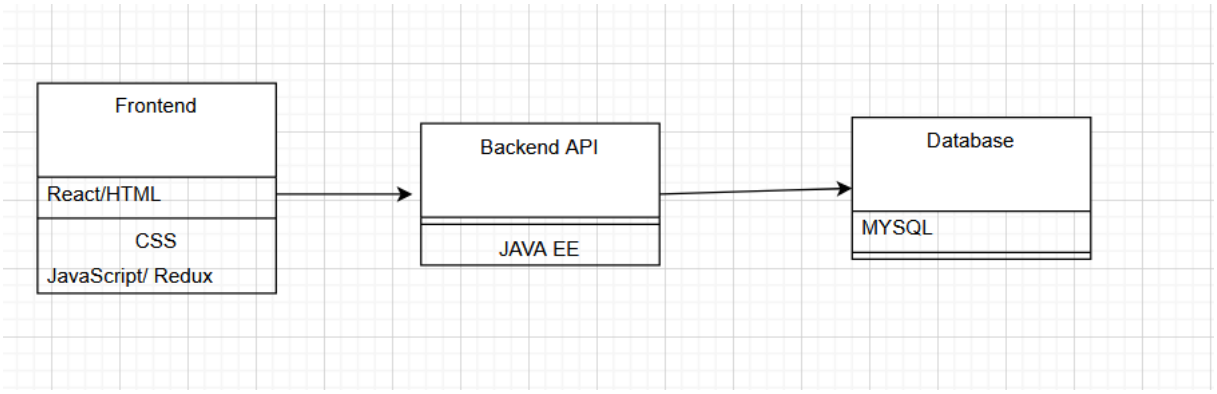


Figure 1 Architecture diagram for theater webapp

In this solution, the frontend of the web application is built using React/HTML/CSS/JavaScript and Router dom/Redux for state management. The frontend communicates with the backend API, which is built using RESTful API for the frontend to access data from the database. The backend API also includes an authentication mechanism to ensure that only authorized users can access the communication platform.
The database used in this solution is MySQL.

To Do Detail Information

1, Create and managing members information
2, Authentication or login
3, application send email to the user
4, find by email
5, create and manage event
6, display event in a calendar mode
7, search event by title
8, display list of emails sent to the member

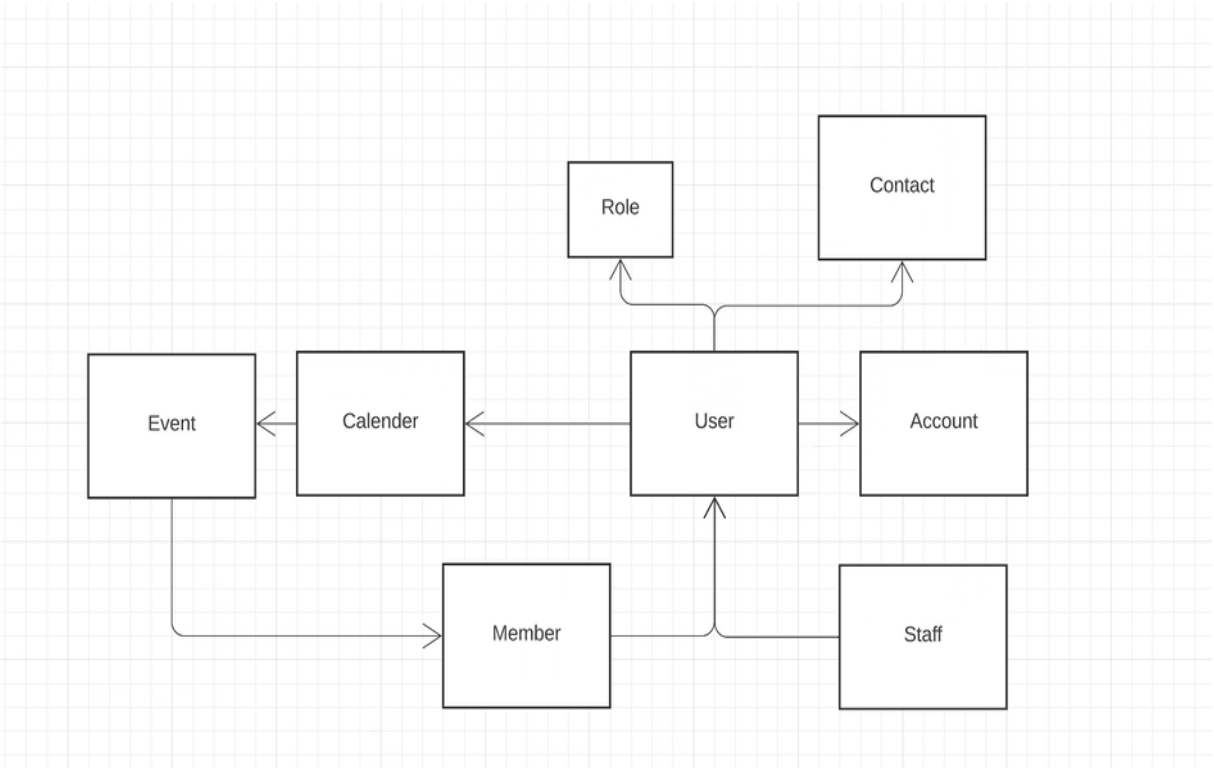


Figure 2 Domain class diagram

More detailed

Member class.

1, Register Member
2, Add Member
3, Remove Member
4, Find Member

Event class.

1, Event Name
2, Start Time
3, End time
4, Location
5 Attendees (add Attendee, remove Attendee, get Attendees)

Calendar class.

private List <Event> events.

The Calendar class has a private instance variable for a list of events.
1, Add Event
2, Remove Event
3, Get Event

Staff class

1, Id
2, First Name
3, Last Name
4, Birthdate
5, Email
6, user

User Class

1, Username
2, Password
3, Role
4, Contact Information

Contact Information Class

1, Id
2, Phone
3, Address
4, City
5, Zip Code

Account Class

1, Username
2, Password

Role Class

1, Role_Admin
2, Role_member

The dependency we are going to use in spring framework.

- Security dependency minimum java 17
- Spring boot Dev Tools
- Spring web
- Spring security
- Lombok
- JDBC API
- Spring data JPA
- H2 Database
- MySQL Driver
- Validation
- Spring boot starter mail

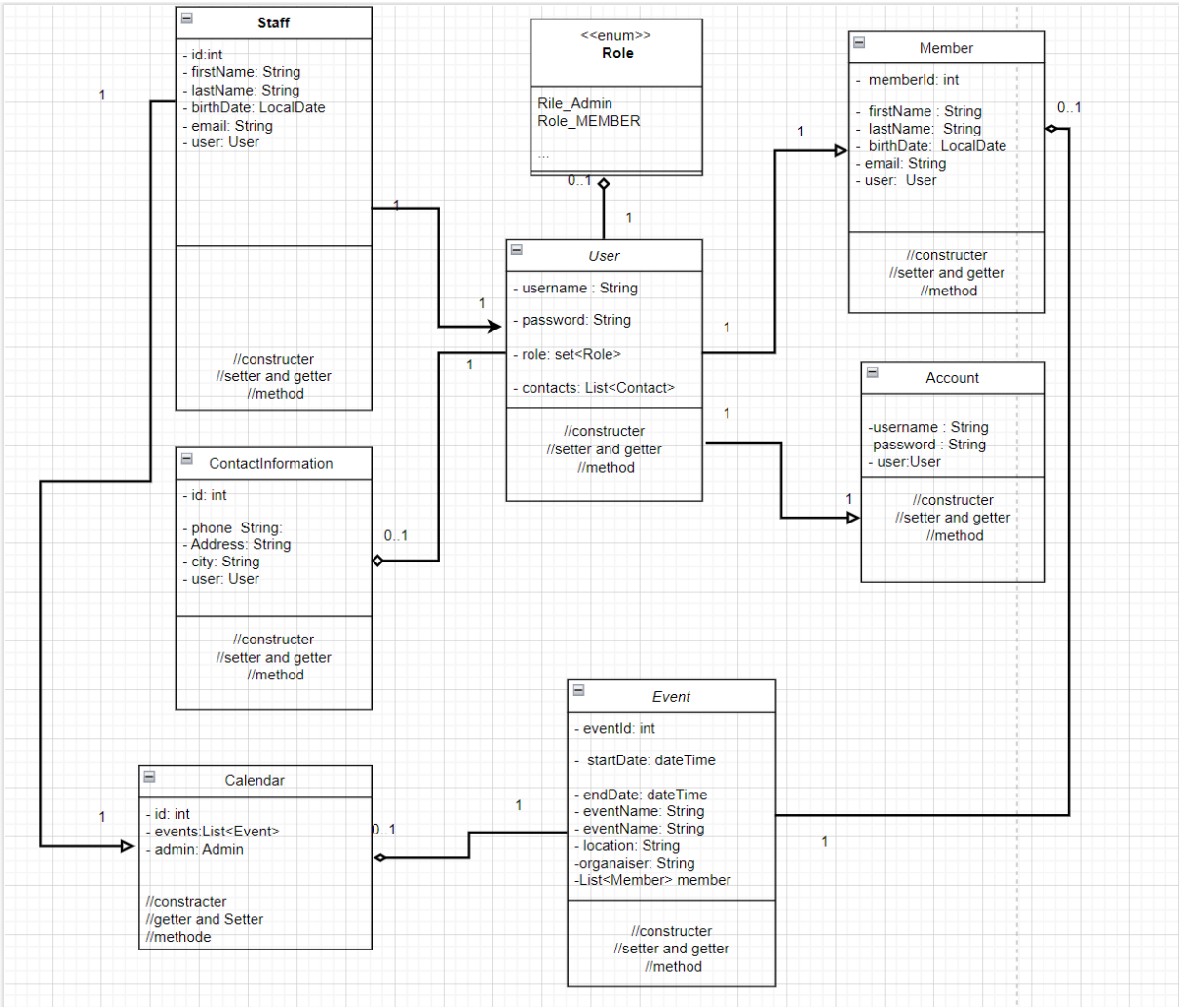
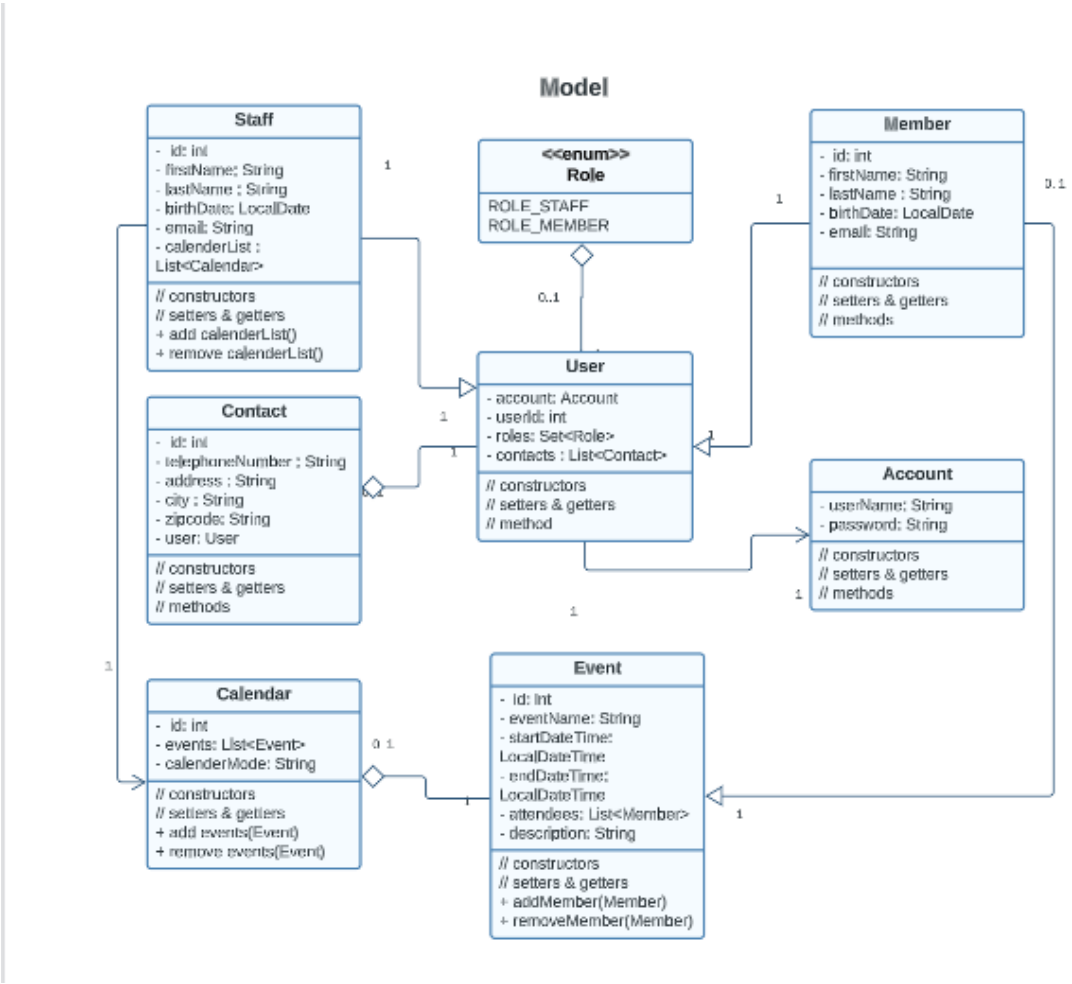
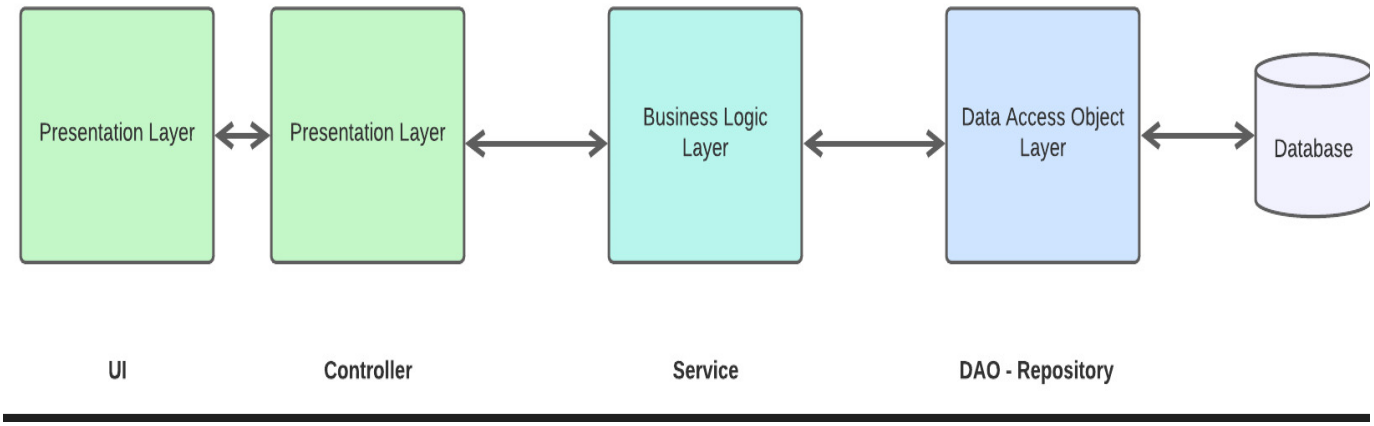


Figure 3 class digram



Credential class to represent the authentication mechanism. The Credential class has a one-to one relationship with the User class and contains the User's login credentials. Contact Information class to represent the Member and Staff's contact information.

Finally, Staff class to represent the staff members who have access to create and manage events. The Event class has a one-to-many relationship with the Staff class, indicating that each event can be managed by one or more staff members. The Member class does not have a direct relationship with the Event class, indicating that members cannot create or manage events.


Presentation Layer
Presentation Layer
Business Logic Layer
Data Access Object Layer
Database
UI
Controller
Service
DAO - Repository