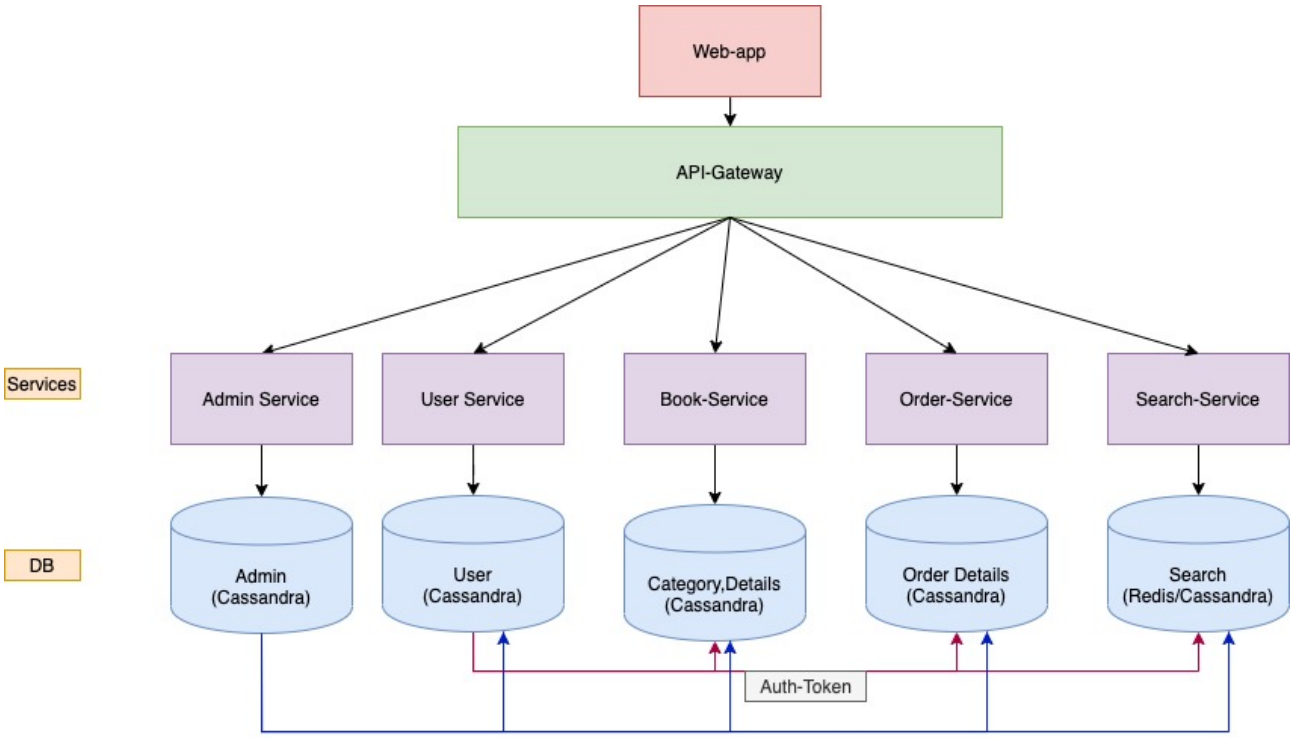
Online Book store Management System



Services

- **Auth-Service:** Responsible for handle authentication process of the system, auth-user retrieve , auth-user store. It connects with User, Role, Permission db tables.
- **Book-Service**: Responsible for book store for all kind users.
- **Order-Service:** Responsible for handling orders which placed by buyers for the products.
- **Search-Service:** Responsible for searching books, users, categories.

Gateway

- **API-Gateway: Netflix Zuul API Gateway** (Load balancing also done)

DB

- **Cassandra**
- **Redis** *(Maybe)*

Service Discovery

- **Eureka *(Host on GitHub repository , endpoints):*** Created on LocalHost.

Auth-Service:

Usage: Oauth2 **(Tried).** Used : Spring Security Authorization filter.

User: UserID *(PK)* ,Username *(Unique)* , Name, DOB, Email, Password, Address, PhoneNo.,
Role: RoleID *(PK)* , RoleName *(Unique)* , Role description.
Permission: PermissionID *(PK)* , PermissionName *(Unique)*

User to Role : *(UserID , RoleID)*
Role to Permission : *(RoleID , PermissionID)*

Orderid.

Book-Service:

Category : CategoryID *(PK)* , Category *(unique)* .
Details : BookID *(PK) ,* Name , Description , Price, Year, ISBN, AuthorName.

Category to Details : *(CategoryID , RoleID)*

Order-Service:

OrderDetails : OrderID *(PK)* , DateOfOrder , UserID, BookID

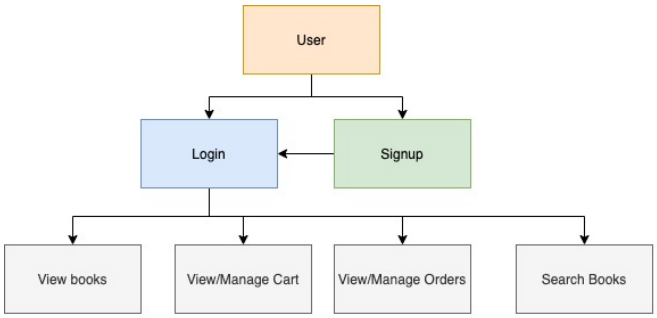
Endpoints:

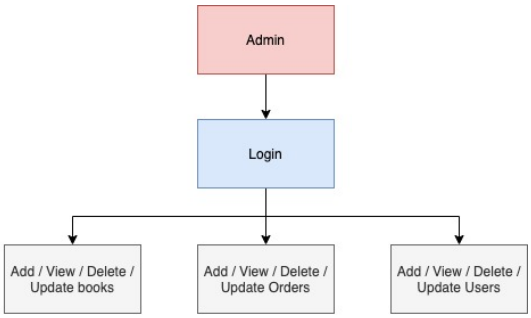
/Users/Login
/Users/Signup
/Users

/Books/Search
/Books/{id}
/Books

/Orders/{id}
/Orders

/Search/Users/ (name)
/Search/Order/ (id)
/Search/Books/ (name)
/Search/Books/Author/ (name)
/Search/Books/Category/ (name)





Front end

- HTML
- CSS
- JS
- React

AWS

Created an account on AWS.

Use of elastic beanstalk to host the WebApp.