**FIS Global Airline Ticket Management System**

Develop an airline ticket management system for FIS Global Airlines. Application must have two types of users roles called Admins and passengers.

Admin is an employee from the FIS Global Airlines, and he is preregistered, his account details need not to be validated. Admin should be able to preform the following functions:

1. Admin can insert a new air flight details like flight\_id, airline\_id, airline\_name, from\_location, to\_location, departure\_time, arrival\_time, duration, total\_seats.
2. Admin can able to modify and delete any air flight details.
3. Admin can able to delete any passenger profile also.

The system should allow passengers to perform the following functionalities:

1. Should be able to register and update them selves, system should read details like profile\_id, password, first\_name, last\_name, address, mobile\_number and email\_id.
2. Should be able to view flight details by providing the from and to locations.

**Design Principles:**

1. Create the application by using Microservices architecture.
2. Use the Feign Client for communication between the micro services.
3. Use Eureka Server for Service Discovery and Ribbon for Load Balancing between the service instances.
4. Create a Single Page Application any of the microservice using Angular and Bootstrap.
5. The angular front-end application should be able to read the data from the backend.
6. Create the appropriate tables in MySQL.

**FIS Video Library Management System**

Develop an Video Library Management System for FIS Video Library Stores. Application must have two types of user roles called Admins and Customers.

Admin is an employee from the FIS Video library store, and he is preregistered, his account details need not to be validated. Admin should be able to preform the following functions:

1. Admin can insert a new movie details like movie\_id, movie\_name, release\_year, language, rating, duration\_in\_minutes, rating, movie\_type, movie\_category, director\_name, lead\_actor\_name1, lead\_actor\_name2, rental\_cost
2. Admin can able to modify and delete any movie details.

The system should allow Customers to perform the following functionalities:

1. Should be able to register and update them selves, system should read details like customer\_id, customer\_name, contact\_no, contact\_address, date\_of\_registration, age.
2. Should be able to search the movies based on movie\_type or lead\_actor\_name1 or rating.

**Design Principles:**

1. Create the application by using Microservices architecture.
2. Use the Feign Client for communication between the micro services.
3. Use Eureka Server for Service Discovery and Ribbon for Load Balancing between the service instances.
4. Create a Single Page Application any of the microservice using Angular and Bootstrap.
5. The angular front-end application should be able to read the data from the backend.

Create the appropriate tables in MySQL.

**FIS Global Bank**

Develop an web application for FIS Global Bank for their customers to provide online functionalities. Application must have two types of user roles called Manager and Customers.

Admin is an employee from the FIS Global Bank, and he is preregistered, his account details need not to be validated. Admin should be able to preform the following functions:

1. Admin can able to create account for a new customer by providing the details like customer\_number, firstname, middlename, lastname, customer\_city, customer\_contactno, occupation, date\_of\_birth.
2. Admin can be able to create details for branch details like branch\_id, branch\_name, branch\_city,
3. Admin can able to create account\_master details like, account\_number, customer\_number, branch\_id, opening\_balance, account\_type, account\_status.
4. Admin can also be able to update and delete customer, branch and account details.

The system should allow Customers to perform the following functionalities:

1. Should be able to allow the customers to view their balance
2. Should be able to create a transaction, by providing the details like, transaction\_number, account\_number, date\_of\_transaction, mode\_of\_transaction, transaction\_type, and transaction\_amount.
3. Should be able to view the transactions based on transaction\_type.

**Design Principles:**

1. Create the application by using Microservices architecture.
2. Use the Feign Client for communication between the micro services.
3. Use Eureka Server for Service Discovery and Ribbon for Load Balancing between the service instances.
4. Create a Single Page Application any of the microservice using Angular and Bootstrap.
5. The angular front-end application should be able to read the data from the backend.

Create the appropriate tables in MySQL.

**FIS Global Airline Ticket Management System (Spring MVC)**

Develop an airline ticket management system for FIS Global Airlines. Application must have two types of users roles called Admins and passengers.

Admin is an employee from the FIS Global Airlines, and he is preregistered, his account details need to be validated.

Once validated, it should re direct to appropriate admin or passengers home page, if not validated, it should be redirected to login page again, by displaying the appropriate error information.

Admin should be able to preform the following functions:

1. Admin can insert a new air flight details like flight\_id, airline\_id, airline\_name, from\_location, to\_location, departure\_time, arrival\_time, duration, total\_seats.
2. Admin can able to modify and delete any air flight details.
3. Admin can able to delete any passenger profile also.

The system should allow passengers to perform the following functionalities:

1. Should be able to register and update them selves, system should read details like profile\_id, password, first\_name, last\_name, address, mobile\_number and email\_id.
2. Should be able to view flight details by providing the from and to locations.

**Design Considerations**

1. Design the application using Spring MVC
2. Connect to databases using the Spring JDBC.
3. User JSP pages for views