**Pharmacy-Management-System**

We will have to accomplish following items in our microservices implementation:

1. Each microservice exposes a set of REST/JSON endpoints for accessing business capabilities.
2. Each microservice implements certain business functions using Spring (Boot) framework.
3. Each microservice stores its own persistent data using H2/MySql/MongoDB database.
4. Each microservice must implement best practices such as, exception handling, Testcases, Static code Analysis and build tools.
5. Microservices are built with Spring Boot, which has an embedded Tomcat server as the HTTP listener.
6. RabbitMQ is used as an external messaging service. Try finding out where it can fit in your case study.
7. The UI, Website must be implemented using React(JavaScript, html and css).

This is Pharmacy Management System case study. In this case study there are two different users 1. admin user and 2. doctor user. Admin is main operator of the pharmacy. He is handling all the functionalities in the pharmacy like view drugs, add/edit drugs, delete drugs, view all the orders placed by doctor user, if order is valid, he can verify that order and can then add that order into picked up orders section if the order has been picked up, view suppliers, add/edit suppliers, delete supplier, view sales report, print sales report, download sales report, etc.

The doctor user will be able to see all the drugs available in the pharmacy and place order according to the requirement.

**Milestone:**

1. Header Templating: It will include Login as a Doctor and Login as Admin buttons on right-side of the page.
2. Header-Routing: Linking different components with each other and able to access each page.
3. Implementing-Angular: Implementing how to use angular to consume JSON.
4. Implementing-Rest-Template: Implementing how to use REST template to consume JSON.
5. Implementing-MongoDB:

* Implementing users collection which provides info about admin user and doctor user.
* Implementing orders collection which provides info about all the orders, verified orders and pickedUp orders.
* Implementing supplier-inventory collection which provides info about suppliers, inventories and sales.

1. Using Node and Express with Angular: Displaying data from MongoDB into React.

**Features:**

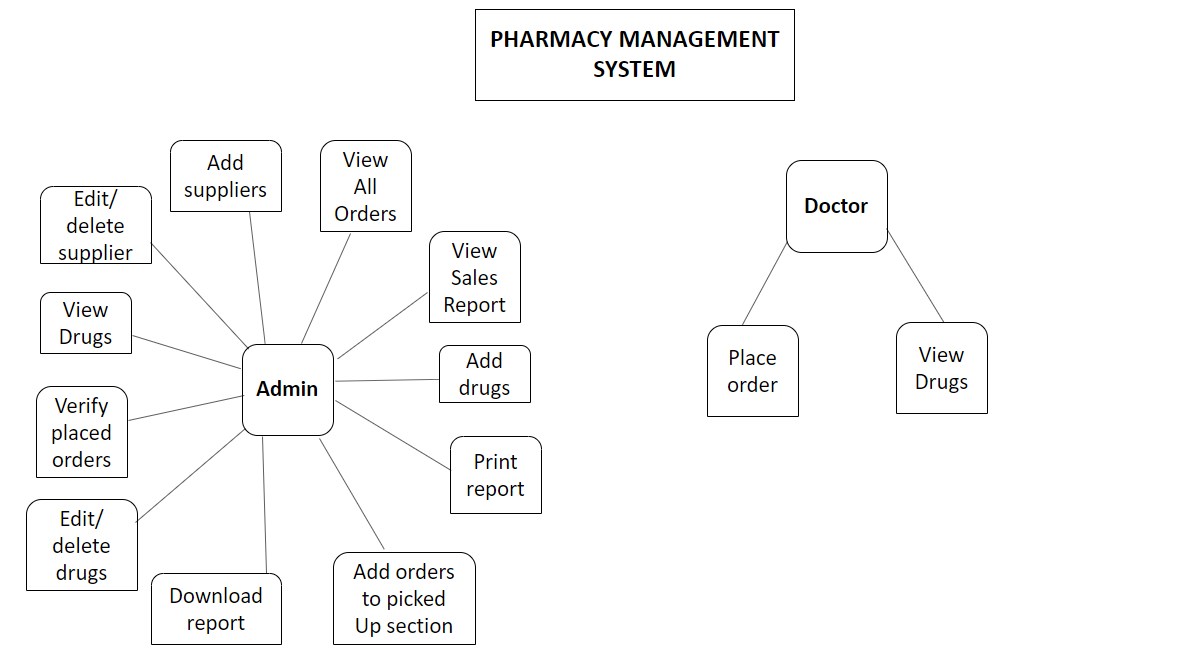
For Admin:

1. Login/Signup: Sign up with basic details name, contact, email and password.
2. View drugs: View and check details of each and every drug available.
3. Add/Edit/Delete drugs.
4. View suppliers: View and check details of each and every supplier.
5. Add/Edit/Delete supplier.
6. View doctor orders: View orders placed by all the doctors and check if the order is valid or not.
7. Verify orders: On validating the order, he can verify the order.
8. Add order to pickedUp section: If the order has been picked up then add that order to pickedUpOrders section.
9. Views Sales Report.
10. Download/Print sales report.

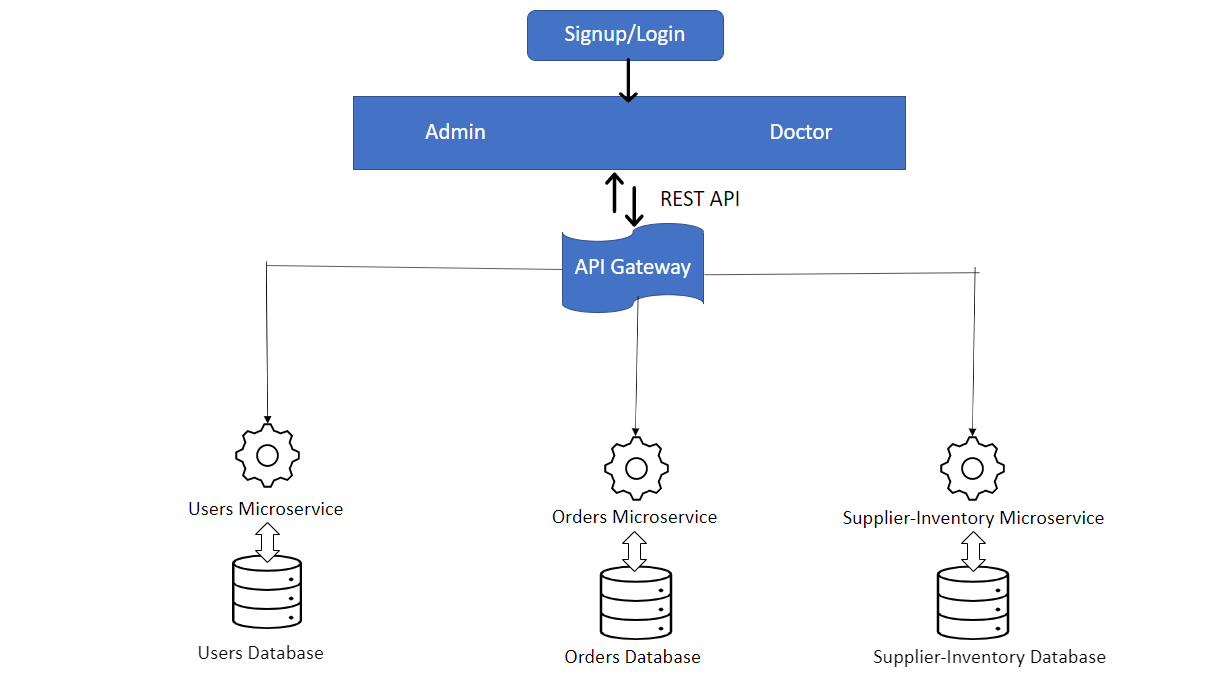
For Doctor user:

1. Login/Signup: Sign up with basic details name, contact, email and password.
2. View drugs: View and check details of each and every drug available.
3. Buy drugs.

**Block Diagram of Pharmacy-Management-System:**



**Architectural Diagram of microservices:**



**Sample Images of Front-end**

