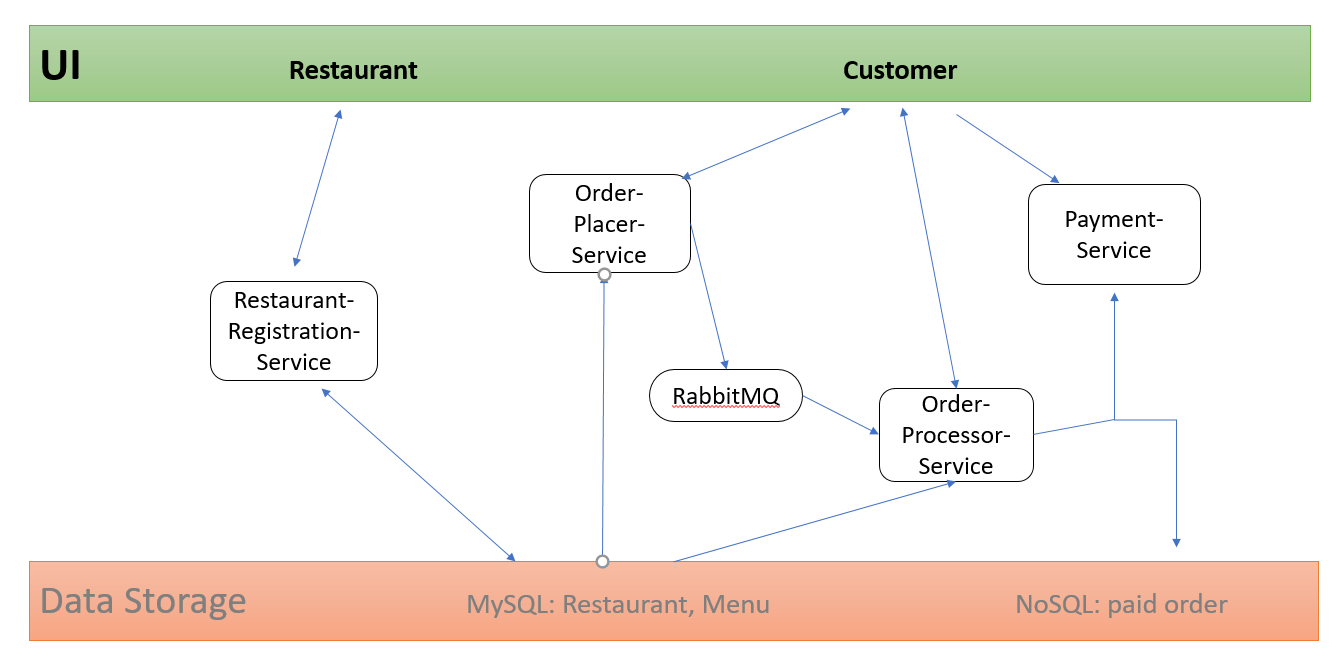
**System Architecture**



**Project Structure:**

1. Restaurant-Registration-Service

This service is to upload/check restaurant and menu information to/from database.

Domain:

Resturant.class: name, address, Menu

Menu.class: <item, price>

One-to-one relationship between Restaurant and Menu.

Rest controller / REST API:

POST: upload restaurant and its menu

GET: search by restaurant name and return its menu

DELTE

Database: mySql

Dependencies:

spring-boot-starter-web, spring-boot-starter-data, maven-plugin

2. Order-Placer-Service

This service is to search restaurant and menu, and place order to Order-Processer-Service.

Domain:

Order.class: customerName, restaurant, deliveryAddress, Items, customerNote

Items.class: name, quantity

REST API:

GET: search by restaurant name and return its menu

POST: place an order with items; send orders to Order-Processer-Service

Database: mySql

Dependencies:

spring-boot-starter-web, spring-boot-starter-data-jpa, maven-plugins, amqp-client

3. Order-Processer-Service

This service is to calculate order price, send payment request to Payment -Service and return payment confirmation to customer/frontend UI.

Domain:

PlacedOrder.class: customerName, restaurant, deliveryAddress, Items, customerNote

Items.class: name, quantity

Service:

PaidOrderDAO.class: Items, prices, orderTime, estimateDeliveryTime

PaymentCalculation.class: Order, totalPrice

REST API:

GET: get placed order

POST: send payment request; send payment confirmation to custmer

Database: mySql, MongoDB

Dependencies:

spring-boot-starter-web, spring-boot-starter-data-jpa, spring-boot-starter-data-mongodb, maven-plugins, amqp-client

4. Payment-Service

This service is for payment.

Domain:

Payment.class: Payer, paymentID, timestamp, items, totalPrice

Payer.class: name, cardNum, expireDate, securityCode

REST API:

POST: provide customer card information

Dependencies:

spring-boot-starter-web, maven-plugins, amqp-client