**There are 2 types of user accounts**(user can choose the type during the registration process):

* Consumer account
* Seller account

There are 5 types of users:

* **Consumer**(can buy products, rate products, add item to shopping cart etc.)
* **Premium consumer**(has functionality of user + gives additional features like free delivery and refund)
* **Seller**(user can sell products, create offers, manage own offers(deactivate change price, change description)
* **Admin**(can manage other users accounts[both consumers and sellers e.g. block account, remove opinion, remove offer, send the email message to any user]
* **Unauthorized user** – can browse the service, check prices, browse products, **but cannot** purchase, leave opinions, grade products

The **consumer** can buy premium for specific time period(the price will different for each of them):

* 1 month
* 3 months
* 6 months
* 12 months

**Options for registration**:

* Using login and password(with submission sent to email address)

**Additional options**:

* Enable two factor authentication(using the email or message on the phone?)(in order to activate it for the first time user need to confirm it also by email

**Search engine**: Elasticsearch

Search: user makes request to search endpoint using filters(price,category[admin defines categories],based on phrase entered by user)

*!!!Integration of inpost api needs the NIP number*

*!!!Seller account on allegro also requires NIP number*

*Mayber find it on pit, nothing happens if I try 😊*

**Reservations**: there is no reservation system but the item can be added to the shopping cart, and will be stored even if you logout. But since it can stay for some time in shopping cart, the status of it can change, and the product becomes unavailable then the user cannot purchase since this item is not available

*Use some apis maybe like twilio?*

**Microservices:**

* Registration 8081
* Authentication
* Email sender
* Api gateway: will contain authorization(jwt token) 8080

…

After user registers it will return short message like email was sent and will send the message to message queue which will be implemented using rabbitmq

**Subsriptions** will be stored in separate table(subscription\_id,user\_id,subscription\_type,start\_date,end\_date etc.)

for now the purpose of registration microservice is to get the data from the user, validate the data and create the instance in the database without sending an email, just return the message that email has been sent.  
  
  
!Also use cookies

app\_user entity should be shared with feign since it will be reused in other microservices  
  
maybe firstly do with user inside the microservice and then refactor

for each microservice security configurations are needed so user can not access the endpoint in microservices directly(without api gateway)

Gateway: 8080  
RegistrationMicroservice: 8081  
Notification: 8082  
Authentication: 8083  
Zipkin : 9411  
eureka: 8761