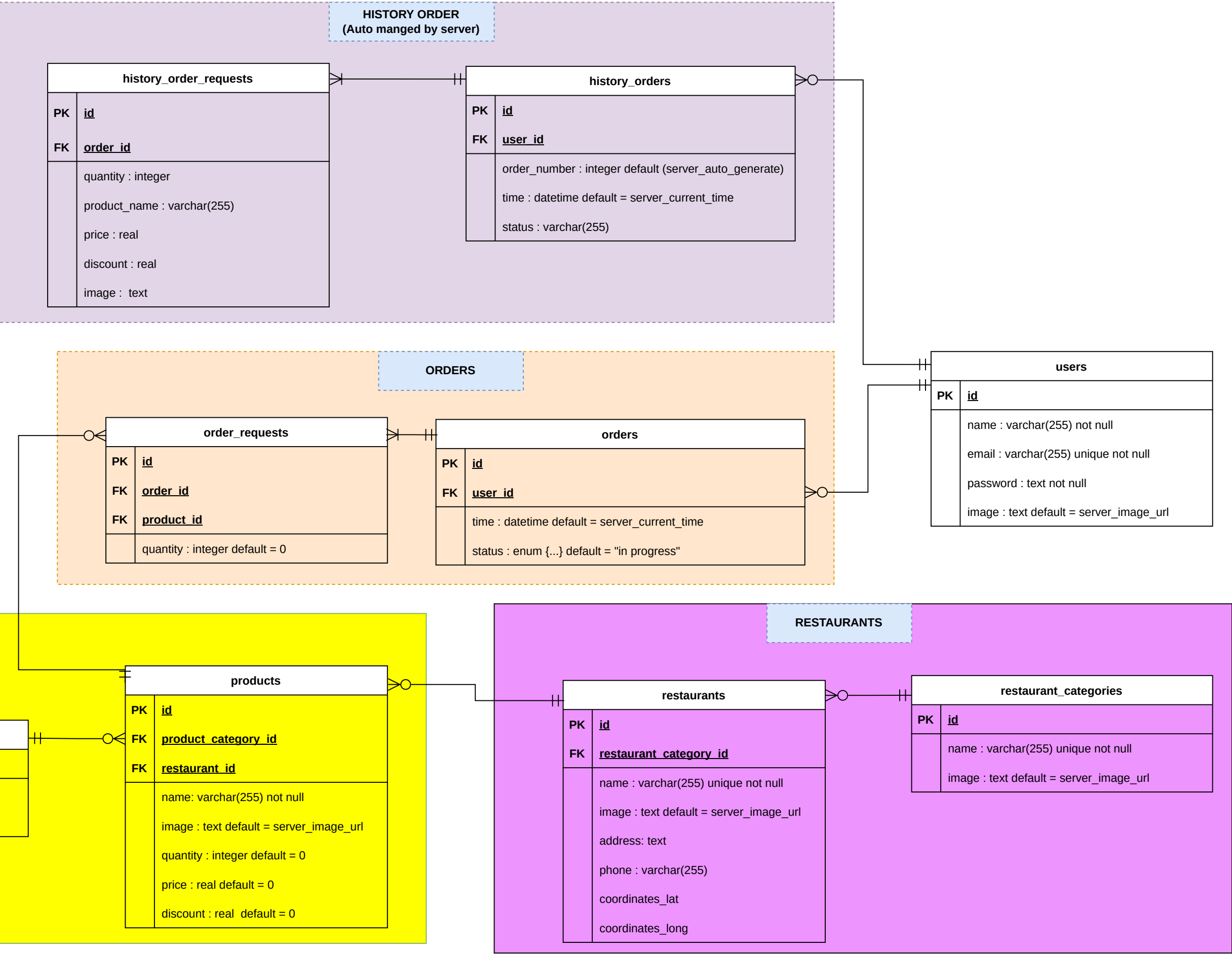


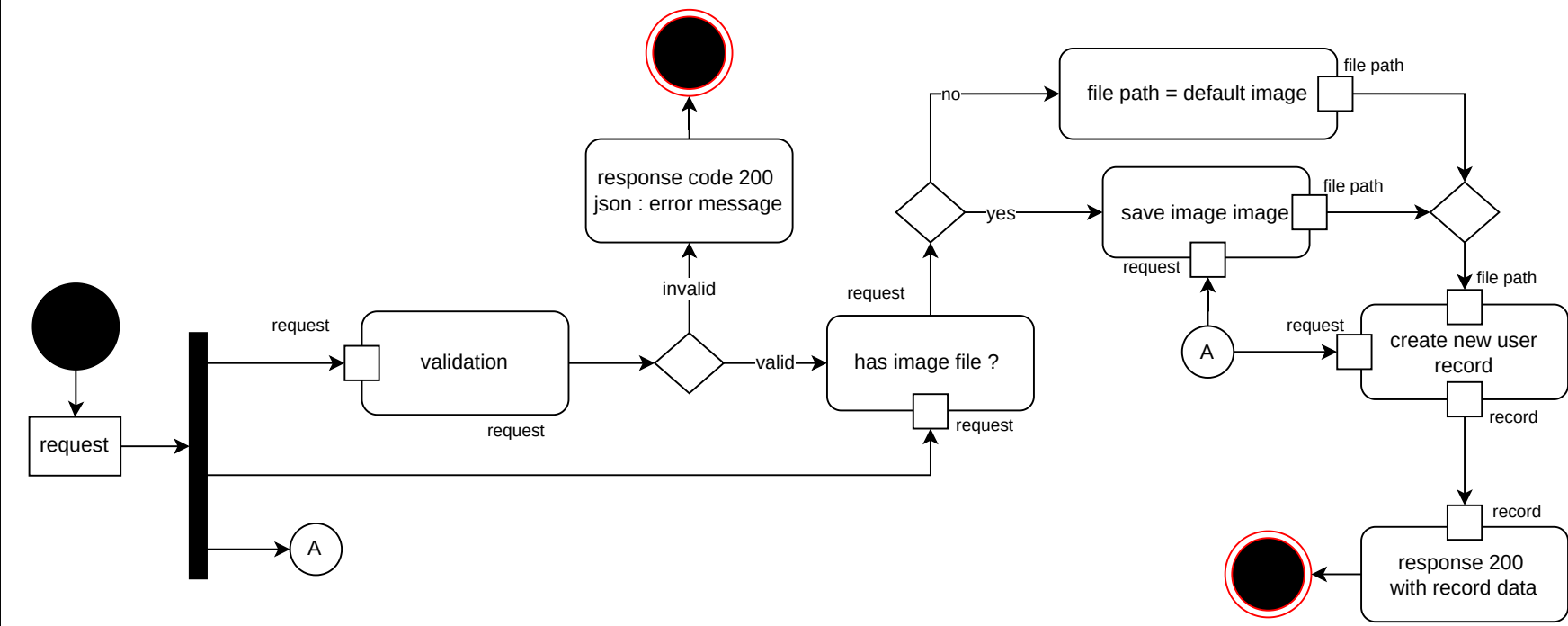
Schema



These actions is do the same behaviour on tables :
(restaurant_categories , restaurant , product_categories, categories)

Act : create new user

<<precondtion>> schema://FQDN/api/user
<<precondition>> HTTP method = POST
<<precondition>> HTTP header content-type = form-data



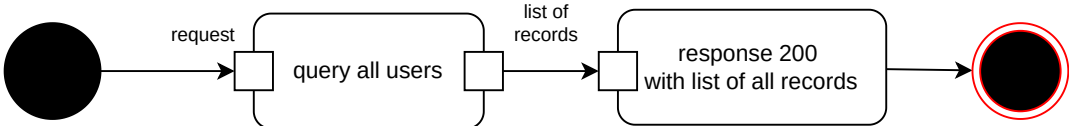
REQUEST
{
 "name" : "user name",
 "email" : "mail@domain.com" ,
 "password" : "123",
 "image" : FILE
}

RESPONSE [ok]
{
 "operation" : true,
 "id" : 1,
 "name" : "user name",
 "email" : "mail@domain.com" ,
 "image" : http://domain.com/storage/image.jpg
}

RESPONSE [fail]
{
 "operation" : false,
 "messages" : "why it false",
 "errors" : ["error 1" , "error 2" , ...]
}

Act : read all users

<<precondtion>> schema://FQDN/api/user
<<precondition>> HTTP method = GET
<<precondition>> HTTP header content-type = json

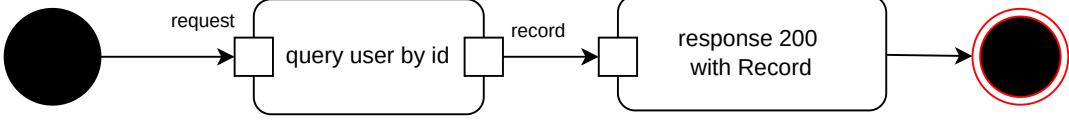


REQUEST
{
}

RESPONSE
[
 {
 "id" : 1,
 "name" : "user name",
 "email" : "mail@domain.com" ,
 "image" : http://domain.com/storage/image.jpg
 }
] 0..*

Act : read single user

<<precondtion>> schema://FQDN/api/user/{id}
<<precondition>> HTTP method = GET
<<precondition>> HTTP header content-type = json

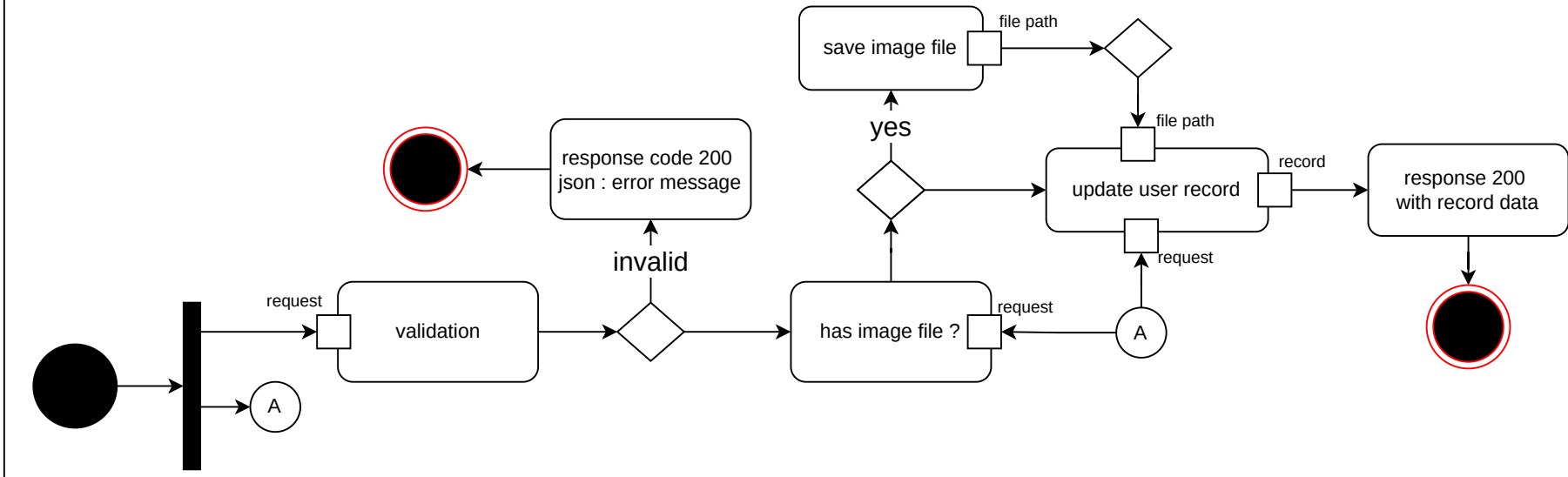


REQUEST
{
 "id" : 1
}

RESPONSE
{
 "id" : 1
 "name" : "user name",
 "email" : "mail@domain.com" ,
 "image" : http://domain.com/storage/image.jpg
} 0..1

Act : update user

<<precondtion>> schema://FQDN/api/user/update/{id}
<<precondition>> HTTP method = POST
<<precondition>> HTTP header content-type = form-data



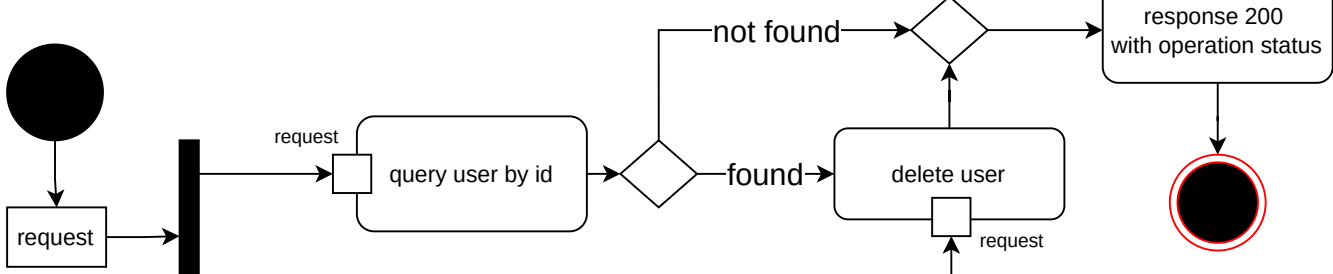
REQUEST
{
 "name" : "optional - new user name",
 "email" : "optional - newemail@domain.com" ,
 "password" : "optional - new password" ,
 "image" : optional - new image FILE
}

RESPONSE [ok]
{
 "operation" : true,
 "id" : 1
 "name" : "new user name",
 "email" : "newmail@domain.com" ,
 "image" : http://domain.com/storage/new-image.jpg
}

RESPONSE [fail]
{
 "operation" : false,
 "messages" : "why it false",
 "errors" : ["error 1" , "error 2" , ...]
}

Act : update user

<<precondtion>> schema://FQDN/api/user/{id}
<<precondition>> HTTP method = DELETE
<<precondition>> HTTP header content-type = json



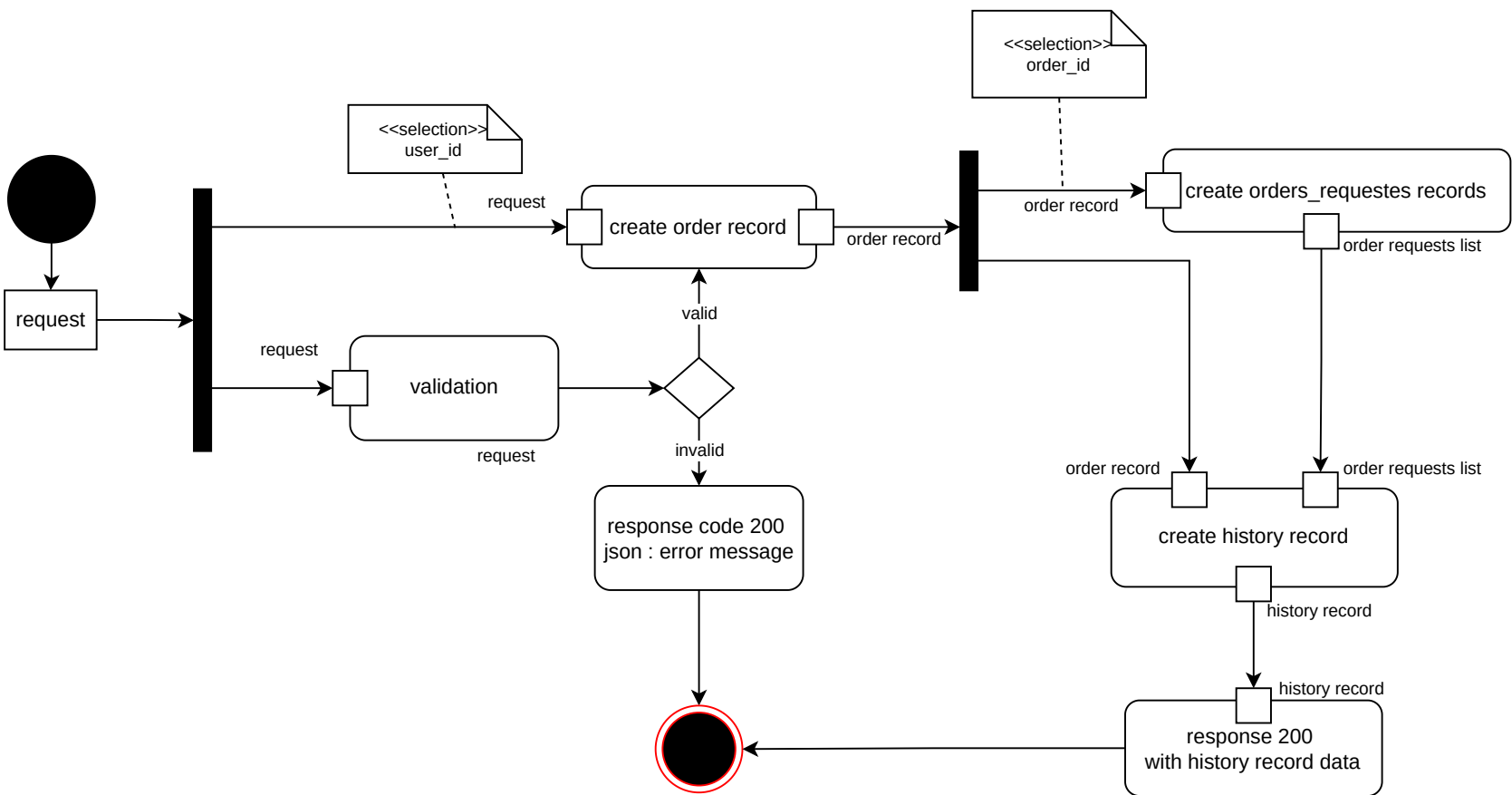
REQUEST
{
}

RESPONSE [ok]
{
 "operation" : true,
}

RESPONSE [fail]
{
 "operation" : false,
 "messages" : "not found",
}

Act : create new order

<<precondtion>> schema://FQDN/api/order
<<precondition>> HTTP method = POST
<<precondition>> HTTP header Authorization Token required
<<precondition>> HTTP header content-type = json



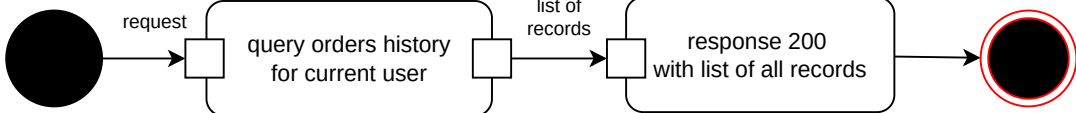
```
REQUEST
{
  "user_id" : 1,
  "products" : [
    {
      "id" : 1 ,
      "quantity " : 2,
    } 1..*
  ]
}

RESPONSE [ok]
{
  "operation" : true,
  "order_id": 1 ,
  "restaurants_id" : 1,
  "restaurants_name" : "Pizza hut",
  "time" : "2023-12-01 14:22:00",
  "total_price" : 600,
  "products": [
    {
      "id": 1,
      "name" : "product name",
      "price" : 200,
      "quantity ": 3,
      "total_price" : 600,
    } 1..*
  ]
}

RESPONSE [fail]
{
  "operation" : false,
  "messages" : "not found",
  "errors" : ["error1" , "error2", ...]
}
```

Act : read all history orders

<<precondtion>> schema://FQDN/api/order/
<<precondition>> HTTP method = GET
<<precondition>> HTTP header Authorization Token required
<<precondition>> HTTP header content-type = json

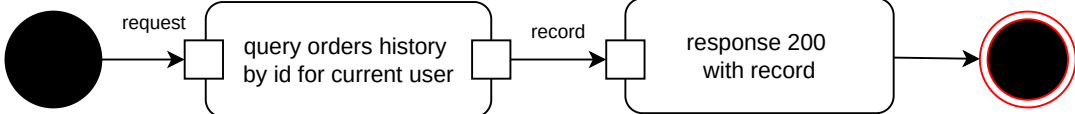


```
REQUEST
{
}

RESPONSE
[
  {
    "order_id": 1 ,
    "restaurants_id" : 1,
    "restaurants_name" : "Pizza hut",
    "time" : "2023-12-01 14:22:00",
    "total_price" : 600,
    "products": [
      {
        "id": 1,
        "name" : "product name",
        "price" : 200,
        "quantity ": 3,
        "total_price" : 600,
      } 1..*
    ]
  } 0..*
]
```

Act : read single history orders

<<precondtion>> schema://FQDN/api/order/{id}
<<precondition>> HTTP method = GET
<<precondition>> HTTP header Authorization Token required
<<precondition>> HTTP header content-type = json

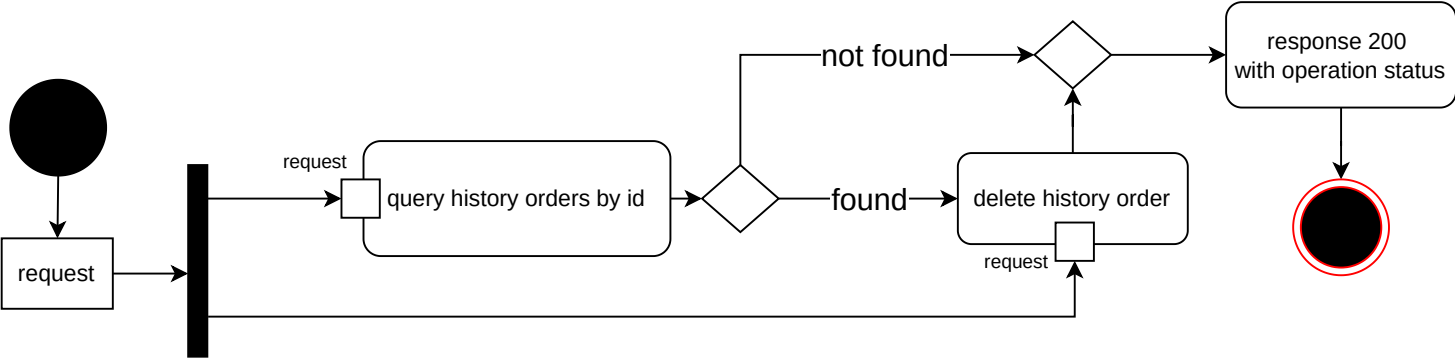


```
REQUEST
{
}

RESPONSE
{
  "order_id": 1 ,
  "restaurants_id" : 1,
  "restaurants_name" : "Pizza hut",
  "time" : "2023-12-01 14:22:00",
  "total_price" : 600,
  "products": [
    {
      "id": 1,
      "name" : "product name",
      "price" : 200,
      "quantity ": 3,
      "total_price" : 600,
    } 1..*
  ]
} 0..*
```

Act : delete history orders

<<precondtion>> schema://FQDN/api/order/{id}
<<precondition>> HTTP method = DELETE
<<precondition>> HTTP header Authorization Token required
<<precondition>> HTTP header content-type = json



```
REQUEST
{
}

RESPONSE [ok]
{
  "operation" : true,
}

RESPONSE [fail]
{
  "operation" : false,
  "messages" : "not found",
}
```