2 Design an app for calling taxi (eg. Uber)

Objects:

Taxi:

Data: Driver, Location, Passengers,

Behaviors: getLocation, isAvailable, acceptRequest,

TaxiClient (Used by Taxi Driver):

Behavior: acceptRequest(), getTaxi()

Passenger:

Data: Name, Telephone, Email, Address, CreditCard

AppClient (Used by Passenger):

Data: passenager, passengerID, network,

Behavior:register, login,

makeRequest, makePayment,completeRequestWithTaxi,

Request (Describe the request form the Passenger):

Data: destination, location, departureTime

BackendService (Include the schedule of taxis to passengers):

Data: taxis, passengers,

Behavior: findBestTaxi, recevieRequest, findTaxiNearBy, notifyClient

CallTaixProcess:

Passenger peter,

AppClient peterClient,

Request peterRequest,

Taxi bestTaxi,

```
    BackendService centralService,

    // Register or Login
    if peter not register in client:
        peterClient.register(peter)

    peterClient.login(peter)

    // Peter makes a request and centralService find a taxi for him
    peterRequest = peterClient.makeRequest()

    When centralService receive the new request:
        centralService.addRequest(request)
        bestTaxi = centralService.findTheBestTaxi(request)
        centerSerivce.notifyClient(peterClient)
        peterClient.completeRequestWithTaxi (bestTaxi)
        peterClient.makePayment()

// How to find the best taxi
SubProcess: findTheBestTaxi
Taxi centralService.findTheBestTaxi(request):
        taxiClients = centralService.findTaxiNearBy(request)
        for taxiClient in taxiClients:
            if taxiClient getTaxi() is available and
                                    taxiClient. acceptRequest(request):
                return taxiClient.getTaxi();
```