

Problem : Design an app calling taxis

Thing(Object)/Entity:

Information(Data)/(State):

Services(Behaviors)/Actions

Thing:

OnlineService:

Data: drivers(list of drivers), users(list of active users)

Behaviors: bookTaxisForUser(), searchNearbyDrivers(), estimateFee()

User:

Data: name, location

Behaviors:

Driver:

Data: name, id

Behaviors: isAvailable, acceptCustomer

LoginToAppAuthService

Sequence of invoking behaviors on objects:

User: user

Driver: driver

OnlineService: onlineService

user.LoginToAppAuthService -> app: authorize

if (authorize is true)

user.inputDestination()-> street, zipcode, ..

onlineService.searchNearbyDrivers() -> user.location:

collection of driver(ordered by nearest location)

Loop each driver:

if driver.isAvailable and

driver.acceptCustomer

onlineService.respondToUser()-> user,

driver, estimateFee()

if user.acceptTrip()

onlineService.bookTaxisForUser()-> user, driver

break

else

continue;

else

continue;

else

user.canNotCallTaxis()