

2. Design an app for calling taxis (e.g. Uber).

Wenchao Ma

/**

* neu-sep18 Assignment1

* 2. Design an app for calling taxis (e.g. Uber).

* @author Wenchao Ma

*/

Problem : Design an app for calling taxis (e.g. Uber).

Customers :

Data: UserName, Passcode, FirstName, Pictures, LastName, Birthday, IDType, IDNumber, Address, PhoneNumber, Email

Behaviors : SignIn, SignOut, InputDestination, Location, SearchForDrivers, FindTaxiLocation, Reviews, SubmitOrder, Cancel, Payments, ViewOrders, Setting, Aboard

Drivers:

Data: Name, Address, PhoneNumber, Email, Pictures, Reviews, CarBrand, CarType, CarNumber

Behaviors: OrderReceiving, Navigation, Drive, Location, ConfirmOrder, Arrival, Wait, Contact

Connections:

Data: IpAddressData, DomainData, LocalIpAddressData

Behaviors: EstablishConnection

App:

Data : Name, PhoneNumber, Map, Customers, Drivers

Behaviors : Register, DeleteUserInfo, SignIn, SignOut, SearchForDrivers, SubmitOrder, Payments, SendNotify, Advertisement, Navigation, Recommendation, SendOrdersToDriver, SendOrdersToCustomer, ReceiveInfo

CreditCard

Data : Number, Name, Company, ValidThru, SecurityCode

DebitCard

Data: Number, Name, Company, ValidThru, SecurityCode

Paypal

Data: UserName, Passcode

CardCompany

Data: Interface, AccountData

Behavior: ReceiveTransaction, AuthorizeTransaction, SendTransactionResult

Sequence of invoking behaviors on Objects:

Design an app for calling taxis (e.g. Uber).

Customers taxiCustomer
Drivers taxiDriver
Connections connection
App taxiApp
CreditCard visaCreditCard
DebitCard visaDebitCard
Paypal paypal
CardCompany visaCardCompany

if connection is established
 if this is a guest and has sign up
 taxiCustomer.SignIn -> taxiCustomer.UserName, taxiCustomer.Passcode,
taxiApp, : signResult
 else if this is a guest and has not sign up
 taxiApp.Register -> taxiCustomer.FirstName, taxiCustomer.LastName,
taxiCustomer.Idtype, taxiCustomer.IdNumber, taxiCustomer.PhoneNumber,
taxiCustomer.Email, visaCreditCard, taxiApp : signResult

if signResult means fail
 taxiCustomer.tryAgainLater

taxiCustomer.Location
taxiCustomer.InputDestination
taxiCustomer.SearchForDrivers -> location, destination, carType, taxiApp :
driverList

Initialize pageNumber = 1
Loop in driverList:
 if taxiCustomer findTheDriver
 taxiDriver = findTheDriver
 break
 if driverList.hasNextPage
 pageNumber = pageNumber + 1
 else if driverList.hasNextPage
 break

taxiCustomer.SubmitOrder -> location, destination, taxiDriver, taxiApp :
orderSubmitConfirmation
if orderSubmitConfirmation send success
 taxiCustomer.wait
 taxiApp.SendOrdersToDriver -> taxiCustomer, location, destination, taxiDriver :

```

orderSendToDriverConfirmation
    if orderSendToDriverConfirmation send success
        taxiApp.wait
        taxiDriver.OrderReceiving
        taxiDriver.ConfirmOrder -> orderInfo, taxiApp : orderConfirmation
        if orderConfirmation equals ok
            taxiApp.sendSuccessToCustomer -> orderInfo, taxiCustomer : orderResult
        else orderConfirmation equals fail
            taxiApp.sendFailToCustomer -> orderInfo, taxiCustomer : orderResult
        else if orderSendToDriverConfirmation send fail
            taxiApp.tryAgainLater
    else if orderSubmitConfirmation send fail
        taxiCustomer.submitAgainLater

    if orderResult equals success
        taxiCustomer.waitForDrivers
        taxiDriver.navigation
        taxiDriver.drive -> taxiCustomer.location
        taxiDriver.arrival -> taxiCustomer.location
        taxiDriver.confirmArrivalToLocationOnApp
        taxiDriver.contactWithCustomer -> taxiCustomer.PhoneNumber :
taxiCustomerResponse
    if taxiCustomerResponse equals no problem
        taxiDriver.wait
    else
        taxiDriver.contactAgain
        taxiApp.sendNotifyToCustomer -> driver has arrived
        taxiCustomer.viewArrivalInfoOnApp
        taxiCustomer.contactWithDriver -> taxiDriver.PhoneNumber :
taxiDriverResponse
    if taxiDriverResponse equals no problem
        taxiCustomer.findTaxiLocation
        taxiCustomer.aboard
    else
        taxiCustomer.contactAgain
        taxiDriver.drive -> taxiCustomer.destination
        taxiDriver.arrival -> taxiCustomer.destination
        taxiDriver.confirmArrivalToDestinationOnApp
        taxiCustomer.confirmArrivalToDestinationOnApp
        taxiApp.receiveConfirmationOfArrival
        taxiApp.payments.customerToPay -> price, taxiCustomer : paymentResponse
        /* send payments info to the app */
        customer.payments -> visaCreditCard, customer.Address, taxiApp:
paymentsInfoConfirmation
    if paymentsInfoConfirmation equals success
        /* app send payments info to card company */
        taxiApp.payments.sendToCardCompany -> visaCreditCard,

```

```
customer.Address, customer.FirstName, customer.LastName, CardCompany :
paymentsConfirmation
    if paymentsConfirmation equals success
        taxiApp.payments.sendToDrivers -> order, taxiDriver : orderResult
        if orderResult equals success
            return success to customer and driver
        else
            taxiApp.trySendToDriverAgainLater
            return success to customer
    else
        tell customer to change payment method
else
    tell customer to submit again
else
    taxiCustomer.SearchForDriversAgain

taxiDriver.review.for.customer -> rating good customer or bad customer on App
taxiCustomer.review.for.driver -> rating good driver or bad driver on App
else if connection is null
    taxiCustomer.tryAgainLater
```