

## 1. Order a hotel online before a trip

Objects and behaviors:

Traveller (Internet Shopper):

Data: Name, Phone, TravellingTime, Destination

Behaviors: order, search, review, compare, cancel

Internet:

Data: Airbnb, Xiecheng,

: Group of Websites (Collection of Websites)

Behaviors: searchHotelWebsites

HotelWebsite:

Data: URL, Product[] products, BankAccount

Behaviors: search, sort, display, compare, placeTheOrder

Hotel:

Data: Price, Address, Size, Brand,

Behavior:

CreditCard:

Data: Number, Name, Company, Expire, SecurityCode

Behavior:

CreditCardCompany:

Data: Name, Address,

Behavior: authorizeTransaction

Sequence of invoking behaviors on Objects:

OrderHotelOnline:

Traveller tom,  
Internet internet,  
ElectronicWebsite xiecheng,  
Hotel hilton,  
Creditcard card,  
CreditCardCompany mastercard,  
ShoppingConfirmation response,

If Internet.isAvailable

Tom.searchInternet -> internet, question: Collection of  
HotelWebsite

pageNumber = 1;

Loop

if tom.findNoPages

break

end

tom.findDesirableWebsiteInAPage -> internet, question,

pageNumber: website

xiecheng = website;

if xiecheng is not empty

break

else

pageNumber = pageNumber + 1

end

End

Xiecheng = website

If xiecheng is not empty or xiecheng!=null

Xiecheng.searchForHotel -> priceRange, address, brand:

Collection of hotel

Hilton = hotel

Tom.orderHotel -> Hilton, creditCard, address, xiecheng:

OrderingConfirmation

Response = orderingConfirmation

Else

Tom.cantOrderHotel

Else

Tom.browsInternetAfterAWhileBack

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

Objects and Behaviors:

TaxisPlatform:

Data: name, phoneNumber, website, appStoreAddress,  
emailAddress, bankAccount

Behavior: findTaxis, connect, authorize, getLocation

Caller:

Data: name, phoneNumber, address, destination

Behavior: loginToTaxisPlatform, callTaxis, pay

TaxiDriver:

Data: name, phoneNumber, location, typeOfVehicle, bankAccount

Behavior: receive, answer, drive, pickup, send

Sequence of invoking Objects with Behaviors

CallTaxi:

Caller tom;

TaxiDriver jerry;

Taxisplatform uber;

CallingConfirmation Response;

Tom.loginToTaxisplatform -> tom, password, uber: authorize

Tom.findTaxis -> address, destination, typeOfVehicle: Collection of  
taxiDrivers

driverNumber = 1

loop

if tom.findsNoTaxis

break

end

else

tom.findDeasirableTaxi -> address, destination,

typeOfVehicle: driver

jerry = driver

driverNumber ++

End

Jerry.answer

Jerry.driveToTheStartAddress

Jerry.pickUp

Jerry.send

Tom.pay -> evaluation: CallingConfirmation

Response = CallingConfirmation

If CallingConfirmationIsGood

Uber.payForDriver

Else

Uber.afterSaleService

### 3. Design a job searching and posting platform

Objects and Behaviors:

Company

Data: name, businessType, emailAddress, address, phoneNumber

Behavior: postJob, cancelJob, connect,

Job

Data: name, type, salary, requirement

Behavior

JobHunter

Data: name, age, address, phoneNumber, emailAddress, skill

Behavior: searchJob, connect

Platform

Data: website, emailAddress, phoneNumber

Behavior: showJob, sortJob, receiveRequeirment

Sequence of Flow

JobPostingandSearching:

Company google

Job SDE

JobHunter tim

Platform indeed

Google.loginToJobHuntPlatform -> name, password, indeed: authorize

If(authorize is true)

    Google.postJob -> SDE: authorize

Else

    End

Tim.loginToJobHuntPlatform -> name, password, indeed: authorize

If(authorize is true)

    Tim.searchJob -> jobType : collection of all the job

    jobNumber = 1

    loop

        if tim.findsNoJob

            break

        end

        else

            tim.findDeasirableJob -> skiils, salary, detail: job

            SDE = job

            jobNumber ++

    End



#### 4. Order food in a restaurant

##### Things:

###### Consumer:

Data: name, phoneNumber

Behavior: search, order, pay

###### Restaurant:

Data: type, address, name, phoneNumber

Behavior: cook, receive

###### Waiter:

Data: name

Behavior: offerMenu, offerBill, receiveMoney, serve

###### Menu:

Data: dishes, price, material

##### Sequence of invoking behaviors on Objects:

###### OrderFoodInARestaurant:

Consumer zhao

Restaurant haidilao

Menu menu

Waiter lee

## OrderingConfirmation response

If restaurant.open

    Zhao.askForMenu -> lee, haidilao: m

    Menu = m

    Zhao.seachInMenu -> requirement, menu: Collection of Food

    foodNumber = 1

    loop

        if zhao.findsNoFood

            break

        end

        else

            zhao.findDesirableFood -> requirement, menu,

            foodnumber: Collection of Desirable food

            foodnumber ++

        end

    haidilao.cook

    lee.offerFood

    zhao.eat

    zhao.askForBill -> collectionOfDesirableFood, lee: bill

    zhao.pay -> bill, creditcard: PayingConfirmation

    response = PayingConfirmation

## 5. Design a course registration platform

### Objects and Behaviors:

#### Student

Data: name, studentID, phoneNumber, emailAddress,

Behavior: login, search, register

#### School

Data: name, campus, address, phnoeNumber

Behavior: offerCourse, receive

#### Course

Data: name, detail, place, teacher, schedule,

Behavior

#### Platform

Data: website

Behavior: receive, offer, register, check

### Sequence of invoking behaviors on Objects:

CourseRegistrationPlatform:

Student lin

School neu

Course java

Platform myneu

RegistrationConfirmation response

Lin.loginToCourseRegistrationPlatform -> name, password, myneu:

authorize

If(authorize is true)

Lin.search -> myneu, typeOfCourse: Collection of All offering courses

Loop

courseNumber = 1

Lin.searchForDesirableCourses -> courses, requirement:

Collection of All Desirable Courses

If(course.isAvailable)

Java = course

Else

Break

courseNumber ++

java.register

lin.logout