Problem: order a hotel online before a trip

Identify Objects and Behaviors:

Thing (Object)/Entity:  
 Information (Data)/(State):

Services (Behaviors)/Actions

Things:

Consumer (Internet Shopper):

Data: Name, Phone

Behaviors: pay, search, review, compare, cancel

Internet

Data: Expedia, Student Universe, Airbnb, Google

Behaviors: searchForHotelWebsites

HotelWebsite

Data : URL, Product, BankAccount

Behaviors : search, sort, display, compare, placeTheOrder

Hotel

Data: Price, Location, Rating, Bed, Refundability Free Internet, Shuttle Bus, Breakfast

Behavior:

CreditCard

Data: Number, Name, Company, Expire Date, Security Code

Behavior:

CreditCardCompany

Behavior: authorizeTransaction

Sequence of invoking behaviors on Objects:

OrderHotelOnInternet:

Consumer Sherry,

Internet internet,

HotelWebsite expedia,

Hotel mariott,

CreditCard card,

CardCompany visa,

OrderConfirmation response,

if Internet.isAvailable

Sherry.searchInInternet -> internet, question: Collection of HotelWebsite

pageNumber=1;

Loop

if Sherry.findsNoPages

break

end

else Sherry.findDesirableWebsiteInAPage -> internet,question,page number:website

expedia = website;

if expedia is not empty

break

else

pageNumber ++

end

End

expedia = website

if expedia is not empty or expedia!=null

expedia.searchForHotel-> Price, Location, Rating, Bed, Refundability Free Internet, Shuttle Bus, Breakfast: collection of hotels

mariott=Hotel

Sherry.orderHotel -> mariott, creditCard, address, expedia:orderConfirmation

response=orderConfirmation

else

Sherry.cantOrderHotel

else

Sherry.browseInternetAfterAWhileBack

Problem 2. Design an app for calling taxis

Objects and Behaviors:

Caller

Data: number, name, location, destination

Behavior: loginToPlatform, requestTaxi, confirmTopay

Driver

Data: location, number, name, carType

Behavior: loggedInToPlatform, receiverRequest, completeTrip, pickup

Platform

Data: name, number, appstoreaddress, website, bankaccount

Behavior: assignTaxi, receiveMoney,customerService

Sequence of Flow -Invoke Objects with Behaviors

Caller sherry

Driver vincent;

Platform uber;

sherry.loginToPlatform->uber:authorize

vincent.loggedInToPlatform->uber:authorize

sherry.requestTaxi->location, carType, destination:uber assign vincent

vincent:receiverRequest-> sherry.location,vincent. location

vincent.pickup

vincent.completeTrip;

if (sherry.confirmTopay)

uber.receiveMoney;

else

uber. customerService;

Problem 3. Design a job searching and posting platform

Objects and Behaviors

Things:

JobSearcher:

Data: Profile, Resume

Behaviors: login,search

Company:

Data: name, address, type, positions

Behaviors: loginto, postPostions, cancelPosition, connect

Job:

Data: type, salary, location

Platform:

Data: website, email

Behaviors: sorting, authorize, showResults,

Sequence of invoking behaviors on Objects:

JobSearcher sherry;

Company google;

Job SDE;

Platform linkedin;

google.loginto->name, password, linkedin:authorize

google.postPositions->SDE: authorize;

sherry.login->name,password, linkedin:authorize

sherry.search-> type :collection of jobs

result=1;

Loop

if sherry.findNoJob

break;

end

else

sherry.findDesirableJob ->salary, location, detail: job;

SDE=job;

result++;

end

Problem 4.Order food in a restaurant

Objects and Behaviors:

restaurant

Data: menu, name, address

Behaviors: cook, receiveOrder

menu

Data: collectionofdishes

waiter

Data: name

Behaviors: giveMenu, sendOrder, serveDish, sendCheck

customer

Data: name, dishes, money, requirement(taste..)

Behaviors: searchOnMenu,order, eat, askCheck, payCheck

Sequence of Flow- invoke Objects with Behaviors

customer sherry;

restaurant dingning;

menu Menu;

waiter vincent;

if (dingning is open)

sherry.askmenu-> vincent,dingning :menu

Menu=menu;

sherry.searchOnMenu->menu, requirement: collectionofdishes

numberofdish=1;

Loop

if sherry.findnodish

break;

end

else

sherry.finddesirablefood->numberofdishes, requirement, money,menu: collection of desirable dishes

numberofdishes++

end

vincent.serveDish

sherry.eat

sherry.askCheck-> menu, collectionofdesirablefood, vincent: check

sherry.payCheck->check, card: paymentconfirmation

output=paymentconfirmation;

Problem 5. Design a course registration platform

Objects and Behaviors:

student

Data: studentId, password

Behaviors: login, loopUpClass, register

platform

Data:website

Behavior: authorize, confirmRegistration

course

Data: course number, semester, subject, remainingseats, location, section

Sequence of Flow

student sherry

platform myneu

course course

sherry.login->myneu:authorize

if(authorize is true)

sherry.lookUpClass->semester, subject:collectionofcourses

coursenumber=1

Loop

if sherry.findnoclass

break

end

else

sherry.finddesirableclass-> section, location: collection of desirable courses

coursenumber ++

end

if (remainingseats>=0)

break

end

else sherry.register->course number:myneu.confirmRegistration

else

loginfail