

## course management system

---

Student:

- data:  
name,studentEmail,password,id
- behavior:  
login,logout,passwordChange,viewCourse,searchCourse,  
viewNotification,viewGrade,askForHelp,evaluateTeacher,evaluateTA,  
submitAssignment,downloadLearningMaterial

Teacher:

- data:  
name,teacherEmail,password,id
- behavior:  
login,logout,passwordChange,viewCourse,sendNotification,viewNotification,  
setAssignmentGrade,setLearningMaterial,askForHelp

TA:

- data:  
name,taEmail,password,id
- behavior:  
login,logout,passwordChange,viewCourse,sendNotification,viewNotification,  
setLearningMaterial,askForHelp,setAssignmentGrade

Admin:

- data:  
name,adminEmail,password,id
- behavior:  
login,logout,passwordchange,sendNotification,viewNotification,createAccount,  
deleteAccount,modifyAccount

Course:

- data:  
courseName,courseCredit,courseTime,teacherName,teacherEmail,taEmail,Grade,classroomNumber,  
courseVolume,studentNumber
- behavior:

Notification:

- data:  
title,subject,content

- behavior:

LearningMaterial:

- data:  
name,subject,fileSize
- behavior:

### Sequence of invoking behaviors on objects

```

1. Student ben
2. Teacher siva
3. TA sun
4. Course iNFO5100
5. Admin administrator
6. Notification news
7. LearningMaterial javaBook
8. //student's behavior
9. if ben.login(id,password)
10. if ben wants to view course information
11.   ben.searchCourse(courseName)
12.   ben.viewCourse(courseName)
13. elseif ben wants to evaluate
14.   ben.evaluteTA(sun)
15.   ben.evaluteTeacher(siva)
16. elseif ben wants to view notification
17.   ben.viewNotification(news)
18. elseif there is a ben's notification
19.   ben.viewGrade(iNFO5100)
20. else if ben wants to download learning Material:
21.   ben.downloadLearningMaterial(javaBook)
22. else if ben wants to submit assignment
23.   ben.submitAssignment(file)
24. else
25.   ben.logout()
26. else
27.   askForHelp(administrator)
28. //teacher's behavior
29. if siva.login(id,passowrd)
30. if siva wants to view course information
31.   siva.viewCourse(iNFO5100)
32. if siva wants to see and send notification
33.   siva.viewNotification(news)
34.   siva.sendNotification(news)
35. if siva wants to grade ben's assignment
36.   siva.setAssgnmentGrade(ben)
37. if siva wants to upload learning material named javaBook
38.   siva.setLearningMaterial(javaBook)
39. else
40.   siva.askForHelp(Administrator)
41. //TA's behavior
42. if sun.login(id,passowrd)
43. if sun wants to view course information
44.   siva.viewCourse(iNFO5100)

```

```
45. if sun wants to see and send notification
46.   sun.viewNotification(news)
47.   sun.sendNotification(news)
48. if sun wants to grade ben's assignment
49.   sun.setAssgnmentGrade(ben)
50. if sun wants to upload learning material named javaBook
51.   sun.setLearningMaterial(javaBook)
52. else
53.   sun.askForHelp(Administrator)
54. //Admin's behavior
55. administrator.login(id,password)
56. administrator.viewNotification(news)
57. if one wants to create account
58.   administrator.createAccount(id,password)
59. if one wants to delete his account
60.   administrator.deleteAccount(id,password)
61. if one wants to modify his account
62.   administrator.modifyAccount(id,pawword)
```

## **pet adoption platform**

---

Adopter:

- data:  
id,password,name,email,phonenumber
- behavior:  
login,logout,setInfo,changePassword,searchpet,adoption,askForHelp

Worker:

- data:  
id,password,name,email,phonenumber
- behavior:  
login,logout,setInfo,changePassword,sendNotification,askForHelp

Admin:

- data:  
id,password,name,email,phonenumber
- behavior:  
login,logout,sendNotification,viewNotification,createAccount,deleteAccount

Pet:

- data:  
name,species,age,sex,healthCondition
- behavior:

### Sequence of invoking behaviors on objects

```
1. Adopter ben
2. Worker lei
3. Admin lucy
4. Pet mew
5. //Adopter's behavior
6. ben.login(id,password)
7. if ben wants to modify his information
8.   ben.setInfo()
9. elseif ben wants to search information about cat
10.  ben.search(cat)
11. if ben wants to adopt 'mew':
12.  ben.adoption(mew)
13. else
14.  ben.logout()
15. else
16.  ben.askForHelp()
17. //Worker's behavior
18. lei.login(id,password)
19. if lei wants to send message on platform about 'mew'
20.  lei.sendNotification(mew)
21. elseif lei wants to modify his information
22.  lei.setInfo()
23. elseif he need administrator's help about platform
24.  lei.askForHelp()
25. else
26.  lei.logout()
27. //Admin's behavior
28. lucy.login(id,password)
29. if she needs to modify users' account
30.  lucy.createAccount(id,name)
31.  lucy.deleteAccount(id,name)
32. else if she needs to send or view notification about platform
33.  lucy.sendNotification(content)
34.  lucy.viewNotification(content)
```

### app to book airline ticket

---

Customer:

- data:  
name,id,password,phoneNumber,email
- behavior:  
login,logout,requestCancelOrder,pay,writeReview,searchTicket,viewTicketDetail

Order:

- data:  
orderNumber,flightNumber,startPoint,destination,flightTime,price

- behavior:

airlineCompany:

- data:  
name,id,password,phoneNumber,email
- behavior:  
login,logout,ProcessOrder,setDiscount

```
1. Customer ben
2. Order beijing2Seattle
3. airlineCompany unionAirlines
4. ben.login(id,password)
5. if ben wants to leave Beijing to Seattleby air:
6.  ben.searchTicket(beijing,seattle)
7.  unionAirlines.setDiscount(beijing,seattle)
8.  ben.viewTicketDetail()
9. if ben decide to buy the ticket
10. ben.requestCancelOrder()
11. ben.pay()
12. unionAirlines.processOrder(beijing2Seattle)
13. unionAirlines.sendMessage('Success')
14. if ben wants to write review after flight
15. ben.writeReview('Nice flight')
```

## course registration platform

---

Student:

- data:  
studentName,studentEmail,studentPassword,studentId
- behavior:  
registerCourse,dropCourse,viewCourse,searchCourse,viewRegisterTutorial

Course:

- data:  
className,teacherName,roomNumber,courseNumber,time,subject,volumn,  
RequiredOrElectiveClass,courseDetail
- behavior:

Administrator:

- data:  
adminName,adminEmail,adminPassword,adminId
- behavior:  
login,logout,createAccount,deleteAccount,modifyAccount,  
creatCourse,deleteCourse,modifyCourseInfo

```
1. Student ben
2. Course iNFO5100
3. Administrator admin
4.
5. ben.login(studentId,studentPassword)
6. if ben needs some help:
7.   ben.viewRegisterTutorial()
8. elseif ben wants to register INFO5100
9.   ben.searchCourse('INFO5100')
10.  ben.viewCourse(iNFO5100)
11.  ben.registerCourse()
12. elseif ben wants to drop INFO5100
13.  ben.dropCourse(iNFO5100)
14. //administrator's behavior
15. admin.login(adminId,adminPassword)
16. if admin needs to create,deleteor modify students account
17.  admin.createAccount(studentID,studentPassword)
18.  admin.deleteAccount(studentID,studentPassword)
19.  admin.modifyAccount(studentID,studentPassword)
20. elseif admin needs to create,deleteor modify one course
21.  admin.createCourse(Course)
22.  admin.deleteCourse(Course)
23.  admin.modifyCourse(Course)
24. else
25.  admin.logout()
```

## Order food in a food delivery app

---

Customer:

- data:  
name,id,password,address,phoneNumber,creditCard
- bahavior:  
login,requestCancelOrder,pay,searchFood,viewFoodDetail,writeReview,modifyInfo

Food:

- data:  
foodName,picture,price,weight
- behavior:

App:

- data:  
orderId,orderTime
- behavior:  
sendReceipt,allocateCourier,checkout,refund

Shop:

- data:  
shopName,phoneNumber,address,FoodName
- behavior:  
receiveReceipt,contactCourier,contactCustomer,makeFood

Courier:

- data:  
name,phoneNumber
- bahavior:  
deliverFood,contactCustomer

```
1. Customer ben
2. Food taco
3. App app
4. Shop mexicanRestaurant
5. CourierBob
6. ben.login(id,password)
7. if ben needs to modify his information like address or something
8.  ben.modifyInfo()
9. if ben wants to eat taco:
10.  ben.searchFood('taco')
11.  ben.viewFoodDetail(taco)
12. if ben wants to order taco from mexicanRestaurant:
13.  ben.requestCancelOrder(taco,mexicanRestaurant)
14.  ben.pay()
15.  app.sendReceipt()
16.  app.allocateCourier()
17.  mexicanRestaurant.receiveRecepit()
18. if workers from mexicanRestaurant needs to contact withCourier:
19.  mexicanRestaurant.contactCourier()
20. if workers from mexicanRestaurant needs to contact with ben bacause of taco out of stock:
21.  mexicanRestaurant.contactCustomer()
22.  ben.RequestCancelOrder(taco,mexicanRestaurant)
23.  app.refund(ben)
24.  mexicanRestaurant.makeFood(taco)
25. Bob.deliverFood(app)
26. Bob.contactCustomer(ben)
27. if ben wants to write review after having taco:
28.  ben.writeReview('Delicious!')
```