1. **Design a course management system (Like Canvas):**

- Student
  - Data: name, loginCredentials
  - Behaviors: login, selectCourse, checkAssignment, completeAssignment, completeQuiz, viewFIles, contact
- Teacher
  - Data: name, emailAddress, loginCredentials
  - Behaviors: login, selectCourse, uploadAssignment, uploadQuiz, uploadFiles, contact, giveGrade,
- Course
  - Data: name, file, student, teacher, assignment, quiz
  - Behavior: getFiles
- Assignment
  - Data: file, dueDate, student, teacher
  - Behavior: getDateOfCompletion
- Quiz
  - Data: file, openTime, dueTime, student, teacher
  - Behavior: getStudentOpenTime, getStudentFinishTime
- System
  - Data: name
  - Behavior: notify

```
Student tina;
Teacher siva;
Course info5100
Assignment assign1;
Quiz quiz1;
System canvas;
tina.login(loginCredentials);
tina.selectCourse(info5100);
tina.checkAssignment(assign1.file);
tina.completeAssignment(assign1);
if (assign1.getDateOfCompletion(tina) later than dueDate)
        assign1.notify(tina, siva);
tina.completeQuiz(quiz1);
if (quiz1.getStudentOpenTime(tina) is earlier than openTime)
        canvas.notify(tina);
else
        if (quiz1.getStudentFinishTime(tina) is later than dueTime)
                canvas.notify(tina, quiz1.teacher);
tina.viewFiles(info5100.getFiles());
tina.contact(info5100.teacher);
siva.login(loginCredentials);
siva.selectCourse(info5100);
```

```
siva.uploadAssignment(assign1.file);
siva.uploadQuiz(quiz1.file);
siva.uploadFile(info5100.file);
siva.contact(tina);
if no special system notify
        siva.giveGrade;
```

2. **Design a pet adoption platform**

- Adoptor
    - Data: name, loginCredentials, phone, creditCard
    - Behavior: login, filter, lookAtPet, contactOwner, locateOwner, buyPet, pickupPet
- Donator
    - Data: name, loginCredentials, phone, creditCard, pet, petHealthProof
    - Behavior: login, selectAssociation, donatePetTo, postPet, cancelPost, acceptPayment
- Pet
    - Data: name, currentOwner, location, breed, sex, picture, availability, petHealthProof
    - Behavior:
- Association
    - Data: name, location, pets, logingCredentials, phone, creditCard
    - Beahvior: login, checkPet, updatePet, acceptDonation, contactAdopter
    -

```
Adopter tina
Association petco
Donator tommy
Donator tom
tommy.login(loginCredentials);
tommy.selectAssociation(petco);
Pet lion = tommy.pet(name, currentOwner, location, breed, sex,picture, availability;
if petco.checkPet(tommy.petHealthProof)
        if petco.acceptDonation(tommy)
                tommy.donatePetTo(Petco);
                petco.updatePet(lion);
Pet fanfan = tom.pet(name, currentOwner, location, breed, sex,picture, availability);
tom.postPet(fanfan);
tina.login(loginCredentials);
tina.filter(donator);
tina.lookAtPet(fanfan);
if fanfan is availiable
        tina.contactOwner(fanfan.currentOwner.phone);
```

```
        if both agree
                tina.buyPet(tina.creditCard, tom.creditCard);
                tina.locate(fanfan.currentOwner);
                tina.pickupPet(fanfan);
                tom.cancelPost();
Adopter amy;
amy.login(loginCredentials);
amy.filter(petco);
amy.lookAt(lion);
if lion.petHealthProof is good
        amy.buyPet(amy.creditCard, petco.creditCard);
        amy.contactOwner(petco);
        amy.locateOwner();
        amy.pickupPet(lion);
```

3. **Design an app to book airline ticket**

- Passenger
    - Data: name, login Credentials, creditCard, contactingInfo, destination, arrival
    - Behavior: login, filter, browseTicket, buyTicket, returnTicket, changeTicket, contactAirline, fly
- Airline company
    - Data: name, tickets, passengers, creditCard
    - Behavior: login, updateTicket, return, change, contactPassenger, updateAirlineInfo
- Ticket
    - Data: number, date, destination, arrival, airline, passenger, seatsLeft
    - Behavior:

```
Passenger tina;
Airline delta;
delta.login(loginCredentials);
ticket seaToChina = delta.updateTicket(number, date, destination, arrival, airline, passenger,
seatsLeft);
tina.login(loginCredentials);
tina.filter(date, destination,arrival);
tina.browseTicket(seaToChina);
if seaToChina.seatsLeft is yes
        tina.buyTicket(seaToChina, tina.creditCard);
        delta.updaeAirlineInfo(number, date, destination, arrival, airline, passenger, seatsLeft);
        delta.contactPassenger(delta.seaToChina.passenger);
        if tina cannot follow the new time
                tina.changTicket(seaToChina);
                delta.change();
```

```
            if tina doesn't like the changed ticket
                tina.returnTicket();
                delta.return(tina.creditCard, delta.creditCard);
        else
            tina.fly();
else
        tina.browseTicket();
```

4. **Design a course registration platform**

- Student
    - Data: loginCredentials, name, year, major, emailAddress
    - Behavior: filterMajor, addCourse, dropCourse, addNotify
- Course
    - Data: name, time, location, limit, description, openToYear, currentPeople
    - Behavior:
- Teacher
    - Data: name, course, loginCredentials
    - Behavior: addCourse
- Platform
    - Data: name
    - Behavior: generateSchedule, closeCourse, notify, openCourse

```
Teacher siva;
Platform neu;
siva.login(siva.loginCredentials);
Course 5100 = siva.addCourse(name, time, location, limit, description, openToYear,
currentPeople);
If (5100.currentPeople >= limit)
        neu.closeCourse;
else
        neu.openCourse;
Student tina;
tina.login(tina.loginCredentials);
tina.filterMajor(tina.major);
if 5100.openToYear == tina.year && 5100.currentPeople < limit
        tina.addCourse(5100);
        5100.currentPeople++;
        if tina wants to drop
            time.dropCourse(5100);
            5100.currentPeople--;
        else
            neu.generateSchedule(5100.name, 5100.time, 5100.location);
else
```

```
tina.addNotify(5100);
if (5100.currentPeople < 5100.limit)
        neu.notify(tina.emailAddress);
```

5. **Order food in a food delivery app (like uber eats)**

- User
    - Data: name, phone, loginCredentials, location, creditCard
    - Behavior: login, openRestaurant, selectDish, enterLocation, checkout
- Order
    - Data: name, number, restaurant, driver
- Restaurant
    - Data: name, loginCredentials, phone, location, menu
    - Behavior: login, uploadMenu, receiveOrder, completeOrder
- Driver
    - Data: name, loginCredentials, phone, location
    - Behavior: login, receiveOrder, pickUpOrder, sendOrder, contactUser
- App
    - Data: user, driver, restaurant, order
    - Behavior: sendOrderToDriver, sendOrderToRestaurant, generateRoutine

```
User tina;
Restaurant chinaFirst;
App uber;
Driver tommy;
chinaFirst.login(chinaFirst.loginCredentials);
chinaFirst.menu = chinaFirst.uploadMenu();
tina.login(tina.loginCredentials);
tina.openRestaurant(chinaFirst);
tina.selectDish();
tina.enterLocation(tina.location);
tina.checkout(creditCard);
Order tinaOrder = new Order(tina, 001, chinaFirst, tommy);
uber.sendOrderToRestaurant(chinaFirst, tinaOrder);
chinaFirst.receiveOrder(tinaOrder);
chinaFirst.completeOrder(tinaOrder);
uber.sendOrderToDriver(tommy, tinaOrder);
tommy.receiveOrder(tinaOrder);
uber.generateRoutine(tommy.location, chinaFirst.location);
tommy.pickUpOrder(tinaOrder);
uber.generateRoutine(chinaFirst.location, tina.location);
tommy.contactUser(tina.phone);
tommy.completeOrder(tinaOrder);
```