Assignment1

Name: Yue Qin
NUID: 001567007

1. Design a course management system (Like Canvas);

Student:
  Data: emailAddress, name, loginCredentials.
  Behaviors: login, prelearn, joinMeeting, watchRecording, upload, writeReview.

Professor:
  Data: loginCredentials, name.
  Behaviors: login, upload, organize, teach, assignTasks, gradeAssignments, optimize.

Course:
  Data: name, materials, livesession, classRecording, assignmnet1, assginment1Score.
  Behaviors: N/A.

ManagementSystem:
  Data: name, reviewSystem.
  Behaviors: inform, demonstrateAssignmentsScore, informOfCourseReview, collectReviews.

OnlineMeetingPlatform:
  Data: name, meetingTime.
  Behaviors: inform.

Pseudo Code:

```
Student melo;
Professor james;
ManagementSystem canvas;
melo.login(loginCredentials);
james.login(loginCredentials);
Course math;
james.upload(math.materials);
if melo hasRegisteredThisCourse;
  canvas.inform(melo.emailAddress);
  melo.prelearn(math.materials);
  OnlineMeetingPlatform zoom;
  james.organize(zoom.meetingTime);
  zoom.inform(melo.emailAddress);
  james.teach(math.liveSession, zoom);
  james.upload(math.classRecording, canvas);
  if melo joinLiveSession
    melo.joinMeeting(zoom.meetingTime, math.liveSession);
  else
    melo.watchRecording(math.classRecording, canvas);

  james.assignTasks(math.assignment1);
  canvas.inform(melo.emailAddress, math.asssignment1);
  melo.upload(math.assignment1, melo);
  james.gradeAssignments (math.assignment1Score, melo)
  canvas.demonstrateAssignmentsScore(math.assignment1Score, melo);
  //The same mode with exam.
  canvas.informOfCourseReview(melo.emailAddress);
  if melo satisfiedWithThisCourse
    melo.writeReview(“AAAAAA”);
    canvas.collectReviews(canvas.reviewSystem, math, james);
  else
    melo.writeReview(“BBBBB”);
    canvas.collectReviews(canvas.reviewSystem, math, james);
    james.optimize(math);
else melo hasNotRegisteredThisCourse
```

2. Design a pet adoption platform;

InformationPublisher:
Data: name, loginCredentials, emailAddress.
Behaviors: login, upload.

PotentialAdopter:
Data: name, loginCredentials.
Behaviors: login, applyForAdoption, furthercommunicate, visitPet, providePublisherInformation, confirmAdoptionSuccess, organizePaperworksSigned, terminateApplication, terminateAdoption.

PetAdoptionPlatform:
Data: name.
Behaviors: verifyInformation, publish, verifyRequirement, providePublisherInformation, confirmAdoptionSuccess, organizePaperworksSigned, terminateApplication, terminateAdoption.

Pet:
Data: name, species, appearance, physicalCondition, adoptionRequirement.
Behaviors: N/A.

Pseudo Code:

```
InformationPublisher melo;
PetAdoptionPlatform soulmate;
melo.login(loginCredentials);
```

```
Pet joy = melo.upload(species, appearance, physicalCondition, 
adoptionRequirement);
soulmate.verifyInformation(joy);
if soulmate approveUploadedInformation;
  soulmate.publish(joy);
  PotentialAdopter anna;
  anna.login(loginCredentials);
  anna.applyForAdoption(joy);
  soulmate.verifyRequirement(anna);
  if anna satisfyTheAdoptionInformation
    soulmate.providePublisherInformation(melo.emailAddress);
    anna.furtherCommunicate(melo);
    anna.visitPet(joy);
    if both anna and melo approve
      soulmate.confirmAdoptionSuccess(joy);
      soulmate.organizePaperworksSigned(melo, anna);
    else
      soulmate.terminateApplication(anna);
  else
    soulmate.terminateApplication(anna);
else
  soulmate.terminateAdoption(joy);
```

3. Design an app to book airline ticket;

Customer:

Data: name, loginCredentials, ID, address, phone, creditCard.

Behaviors: login, input, butTicket, requestCancelOrder, prepareToGetAboard, choosePreferredSolution, completeFlight.

AirlineTicketApp:

Data: name.

Behaviors: refer, provide, checkout, transactProcedures, refund, remindNearDepartureTime, offerSolutionsAndGuides, suggestChangePlan.

FlightNumber:

Data: name, details.

Behaviors: N/A.

TravelPlan:

Data: name, departureTime, departurePlace, destination.

Behaviors: N/A.

Pseudo Code:

```
Customer melo;
AirlineTicketApp wings;
melo.login(loginCredentials);
TraverlPlan nirvana = melo.input(departureTime, departurePlace,
destination);
FlightNumber ca666, lh777 = wings.refer(nirvana);
wings.provide(ca666.details, lh777.details);
if melo chooseOneFlightAfterComparation
  melo.buyTicket(ca666);
  wings.checkout(melo.ID, melo.address, melo.phone, melo.creditCard);
  wings.transactProcedures(ca666,melo);
  if melo changesHisMind
    melo.requestCancelOrder(ca666);
    wings.refund(ca666, melo);
  else
    wings.remindNearDepartureTime(melo);
    melo.prepareToGetAboard(ca666);
    if thereIsAnyDelay
      wings.inform(melo);
      wings.offerSolutionsAndGuides(ca666, melo);
      melo.choosePreferredSolution(ca666, wings);
    else
      melo.completeFlight(ca666);
else
  wings.suggestChangePlan(melo);
```

4. Design a course registration platform;

Student:

Data: name, loginCredentials.

Behaviors: login, input, applyForRegistration.

Course:

Data: name, semester, major, details.

Behaviors: N/A.

RegistrationPlatform:

Data: name.

Behaviors: provide, checkSeatsAvailable, checkTimeConflict, approveRegistration, informRegistrationNotAvailable, offerWaitlistOption, putInWaitlist,

Pseudo Code:

```
Student melo;
```

```
RegistrationPlatform aspiration;
melo.login(loginCredentials);
Course calculus, algebra, probability = melo.input(semester, major);
aspiration.provide(calculus.details, algebra.details, probability.details);
melo.applyForRegistration(calculus);
aspiration.checkSeatsAvailable(calculus);
if seatsAvailable
  aspiration.checkTimeConflict(melo);
  if noTimeConflict
    aspiration.approveRegistration(calculus, melo);
  else
    asporation.informRegistrationNotAvailable(calculus, melo);
else
  aspiration.offerWaitlistOption(calculus, melo);
  if melo approveToStayInWaitlist
    aspiration.putInWaitlist(calculus, melo);
  else
    aspiration.informRegistrationNotAvailable(calculus, melo);
```

5. Order food in a food delivery app;

Customer:

Data: name, loginCredentials, address, phone, account.

Behaviors: login, search, evaluate, payForOrder, writeComment, chooseAgain.

FoodDeliveryApp:

Data: name.

Behaviors: checkout, allocateCourier, sendReceipt, sendFeedback, refund.

Food:

Data: name, foodType, falvor, deliveryTime, price, comments.

Behaviors: N/A.

Restaurant:

Data: name, style, deliveryTime, comments.

Behaviors: acceptOrder, contactForExplanation.

Courier:

Data: name.

Behaviors: contactCustomer, deliverFoodPackage.

Pseudo Code:

```
Customer melo;
FoodDeliveryApp hunger;
```

```
melo.login(loginCredentials);
Restaurant santorini = melo.search(style, deliveryTime, comments);
Food spaghetti = melo.evaluate(foodType, flavor, deliveryTime, price,
comments);
melo.payForOrder(spaghetti);
hunger.checkout(melo.address, melo.phone, melo.account);
hunger.sendReceipt(melo.phone);
if ingredientIsSufficient
  santorini.acceptOrder(spaghetti, melo);
  Courier kiyan = hunger.allocateCourier;
  Kiyan.contactCustomer(melo);
  Kiyan.deliverFoodPackage(spaghetti, melo.address);
  if melo satisfiedWithTheFood
    melo.writeComment(“AAAAA”);
  else
    melo.writeComment(“BBBBB”);
    hunger.sendFeedback(melo);
else
  santorini.contactForExplanation(melo);
  hunger.refund(melo);
  melo.chooseAgain;
```