

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

### List of objects and behaviour

Student	Data	studentName,studentEmailAddress, studentPhoneNumber, studentAddress,studentLoginCredentials
	Behaviour	studentRegisterToApp,studentLoginToApp,student Dashboard,studentCourses,studentGrades, studentSyllabus,search,studentCourseModules
App	Data	appCourseList, appDashboard, appToDoList,appStudentProfile
	Behaviour	NA
Dashboard	Data	dashboardCourseAssigned, dashboardToDoList, dashboardAnnouncement
	Behaviour	NA
Courses	Data	courseID, courseName, courseGrades, courseSyllabus,courseAssigments,courseModules
	Behaviour	NA
ToDoList	Data	toDoItemNumber, toDoDueDate
	Behaviour	NA
Help	Data	helplineNumber
	Bahaviour	AccessCanvaGuide,submitTicketToCanvas

```
Student meena
App CMS
Courses AED
meena.studentRegisterToApp(studentName, studentEmailAdress,
studentPhoneNumber);
meena.studentLoginToApp(studentLoginCredentials);
```

```
//To look what are the courses meena has been registered , required  
orientation, toDoList
```

```
meena.studentDashboard();  
meena.studentToDoList(toDoItemNumber,toDoDueDate);  
//To look for courses in details  
    Courses AED=meena.search(courseID, couseName);  
//To look for Grades  
    meena.studentGrades(AED);  
//To check for syllabus  
    meena.studentSyllabus(AED);  
//To check for assignments  
    meena.studentCourseAssignments(AED);  
    meena.studentCourseModules(AED);  
//To check helplineNumber  
    meena.canvasHelplineNumbere(Help.helplineNumber);
```

## 2. Design a pet adoption platform

### List of objects and behaviour

<b>Customer</b>	<b>Data</b>	customerFullName,customerEmailAddress, customerPhoneNumber,customerAddress,customerLogin Credentials,customerCreditCardDetails
	<b>Behaviour</b>	registerToPetApp,loginToPetApp,searchPet,submitAdoptionApplication,payFee
<b>PetApp</b>	<b>Data</b>	listOfPets, location, applicationStatus
	<b>Behaviour</b>	acceptAdoptionApplication, notifyInspectionOfficer,applicationStatus,generateBill,send BillToCustomerEmailAddress,sendDenialMailToCustomer.
<b>Pet</b>	<b>Data</b>	petType, petBreed, petColor,petSize, petAge, petCost
	<b>Behaviour</b>	soundProduced, Kidfriendliness
<b>Inspection Officer</b>	<b>Data</b>	offerName, offierPhoneNumber, officerEmailAdress
	<b>Behaviour</b>	recieveNotificationFromApp,visitCustomerLocation, updateApplicationStatus,doInspection,updateApplicationStatus

Customer deepak  
PetApp myPet  
Pet dog  
InspectionOfficer mohan

#Flow of Application

deepak.registerToPetApp(customerFullName,  
customerEmailAddress,customerPhoneNumber)

deepak.loginToPetApp(customerLoginCredentials)

dog = deepak.searchPet(petType petBreed, petColor, petSize, petAge, petCost)

deepak.submitAdoptionApplication(dog, customerFullName, customerEmailAddress,  
customerPhoneNumber, customerAddress)

myPet.acceptAdoptionApplication(dog, customerFullName, customerEmailAddress,  
customerPhoneNumber, customerAddress)

myPet.notifyInspectionOfficer(customerFullName, customerPhoneNumber, customerAddress)

myPet.applicationStatus = inspectionPending

mohan = receiveNotificationFromApp(customerFullName, customerEmailAddress,  
customerPhoneNumber, customerAddress)

mohan.doInspection()

If mohan is satisfied

    myPet.applicationStatus = mohan.updateApplicationStatus(APPROVED)

else

    myPet.applicationStatus = mohan.updateApplicationStatus(REJECTED)

if(myPet.applicationStatus = "APPROVED")

    deepak.payFee(creditCardCredentials)

```

myPet.generateBill(customerFullName, customerAddress, customerEmailAddress,
customerPhoneNumber)
myPet.sendBillToCustomerEmailAddress(customerEmailAddress)
else
myPet.sendDenialMailToCustomer(customerEmailAddress)

```

### 3. Design an app to book airline tickets.

#### List of objects and behaviour

<b>Customer</b>	<b>Data</b>	customerFullName,customerEmailAddress,customerP honeNumber, customerAddress,customerLoginCredentials, customerCreditCard
	<b>Behaviour</b>	registerApp,loginApp, search, applyDiscount, bookTicket, cancelTicket, writeReviews,requestForRefund, complaint
<b>App</b>	<b>Data</b>	airlinesList,sourceLocation,destinationLocation,journey Cost,journeyDate,discountCoupon
	<b>Behaviour</b>	recieveCustomerBooking, genrateBill, sendBillToCustomerMail, notifyAirline,refund,provideDiscount
<b>Airline</b>	<b>Data</b>	airLineType, airLineNumber, airSourceLocation,airlineDestinationLocation,airlineSe atAvailability
	<b>Behaviour</b>	recieveBookingNotification, sendInvoiceToCustomer

Customer tina  
App makeTrip  
Airline indigo

### ##Flow of Application

```
tina.registerApp(customerFullName,customerEmailAddress,customerPhoneNumber,  
customerAddress);
```

```
tina.loginApp(customerLoginCredentials);
```

```
Airline flight=
```

```
    tina.search(airLineType,journeyDate,sourceLocation,destinationLocation,  
,journeyDate, airlineSeatAvailability);
```

If flight available check if discount available

If discountCoupon available

```
    tina.applyDiscount(flight);
```

```
    makeTrip.provideDiscount(flight);
```

```
tina.bookTicket(flight,customerCreditCard);
```

```
makeTrip.recieveCustomerBooking(flight)
```

```
makeTrip.genrateBill(customerFullName,customerEmailAddress);
```

```
makeTrip.sendBillToCustomerMail(customerFullName,customerEmailAddress);
```

```
makeTrip.notifyAirline(customerFullName,bookingReferenceID);
```

```
indigo.recieveBookingNotification(customerFullName,customerEmailAddress,
```

```
makeTrip.genrateBill(customerFullName,customerEmailAddress));
```

```
indigo.sendInvoiceToCustomer(customerEmailAddress);
```

If(tina changes her mind about journey)

```
    tina.cancelTicket(bookingReferenceID);
```

```
    tina.requestForRefund(bookingReferenceID);
```

Else

```
    flight not available;
```

```

If tina liked the journey and services
    tina.writeReview("*****");
else
    tina.writeReview("00000");
    tina.complaint("reason");

```

## 4. Design a course registration platform

### List of objects and behaviour

Courses	Data	courseID,courseName,courseProfessor,courseDeliveryMent hod,courseFee,seatAvailabiltly
	Behaviour	NA
Student	Data	studentFullName,studentEmailAddress,studentPhoneNumbe r,studentAddress,studentLoginCredentials,studentCreditCard
	Behaviour	registerToCRPapp, loginToApp, registerCourse,cancelRegistration,requestForNewStudentBill
CRPap p	Data	listOfCourse,courseFee,studentBill
	Behaviour	sendConfirmationMailToStudent,generateBill,addCourseFee ToStudentBill,sendStudentBillToStudent,notifyProfessor,prov ideNewBilltoStudent,notifyProfessorStudentDeregistered
Profess or	Data	professorName,professorCourse,professorEmailAddress, professorPhoneNumber,
	Behaviour	receiveNotificationFromCRPapp,sendCourseDetailsToStude nt,removeStudentFromClass

Student deepika  
Course AED  
CourseRegistrationApp CRPapp  
Professor Siva

```
deepika.registerToCRPApp(studentFullName,studentEmailAddress,studentPhoneNumber);  
deepika.loginToApp(studentLoginCredentials);
```

```
Courses course=deepika.search(courseID,courseName,courseProfessor,  
    courseDeliveryMenthod,courseFee);
```

Check if seats available in respective course

If seats available

```
deepika.registerCourse(course);  
CRPapp.sendConfirmationMailToStudent(studentFullName,studentEmailAddress);  
CRPapp.generateBill(course);  
CRPapp.addCourseFeeToStudentBill(studentBill, couseID,CourseFee);  
CRPapp.sendStudentBillToStudent(studentFullName,studentEmailAddress);  
CRPapp.notifyProfessor(profesorFullName, professorEmailAddress);  
siva.receiveNotificationFromCRPapp(studentFullname,studentEmailAddress);  
siva.sendCourseDetailsToStudent(studentFullName,studentEmailAddress);
```

If deepika changes mind about course and want to deregister course

```
deepika.cancelRegistration(course);  
deepika.requestForNewStudentBill(studentFullName,studentEmailAddress);  
CRPapp. provideNewBilltoStudent(studentFullname,studentEmailAddress);  
CRPapp.notifyProfessorStudentDeregistered(studentFullName, studentEmailAddress);  
siva.removeStudentFromClass(studentFullName,studentEmailAddress);
```



## 5. Order food in a food delivery app.(Like Uber Eats)

### List of objects and behaviour

Customer	Data	customerName,customerEmailAddress, customerPhoneNumber, customerAddress, customerLoginCredentials,customerCreditCard
	Behaviour	RegistrationToApiFood, LoginToApp,SearchRestaurant,orderFood,cancelOrder,requestRefund,writeReview,writeCompliant,requestRefund
Restaurant	Data	restaurantName,restaurantMenu,restaurantDish,restaurantDishCost,restaurantAddress,restaurantPhoneNumber,restaurantAccountNumber
	Behaviour	
DeliveryBoy	Data	dbName,dbAddress,dbPhoneNumber,
	Behaviour	dbAcceptPickupOrder,dbDriveToResturant,dbPickupOrder,dbDriveToCustomer,dbContactCustomer,dbDeliverOrder
Dish	Data	dishType,dishRestaurant,dishCost, dishTaste
	Behaviour	NA
App	Data	appRestaurantList,appRestaurantMenu,appOrderTime,appOrderDeliveryTime
	Behaviour	generateBill,checkout,sendReceipt,refund,notifyDeliveryBoy,notifyRestaurant

Customer deepika  
FoodDeliveryApp ApiFood  
Restaurant SarvanaBhavan  
Dish Dosa  
DeliveryBoy Mohan

```
deepika.cRegistrationToApiFood(customerName,customerEmailAddress, customerPhoneNumber);  
deepika.cLoginToApp(customerLoginCredentials);  
Dish dosa=deepika.search(restaurantName,restaurantMenu,restaurantDish,restaurantDishCost);
```

If restaurant listed in ApiFood is open

```
    deepika.orderFood(dosa);  
    ApiFood.generateBill(dosa);  
    ApiFood.checkout(customerAddress,customerPhoneNumber,customerCreditCard);  
    ApiFood.sendReciept(customerEmailAddress);
```

If deepika changes her decision of ordering food

```
    deepika.cCancelOrder(restaurantName,dosa);  
    deepika.requestRefund(customerOrderId)  
    ApiFood.refund(dosa, deepika);
```

Else

```
    Apifood.notifyDeliveryBoy(restaurantName, dosa,restaurantAddress,  
customerAddress);
```

```
    Apifood.notifyRestaurant(appOrderTime,  
        appOrderDeliveryTime,dbName);
```

DeliveryBoy

```
mohan=dbAcceptPickupOrder(restaurantName,dosa,restaurantAddress,customerAddress);  
mohan.dbDriveToResturant(restaurantAddress,restauranPhoneNumber);  
mohan.dbPickupOrder(restaurantName,Dosa,restaurantAddress);  
mohan.dbDriveToCustomer(customerAddress,customerPhoneNumber);  
mohan.dbContactCustomer(deepika.customerPhoneNumber);  
Mohan.dbDeliverOrder(deepika.customerName, deepika.customerAddress);
```

If deepika liked dish

```
    deepika.cWriteReview('*****');
```

Else

```
    deepika.cWriteReview('0000');
```

```
deepika.cWriteCompliant("food was delivered in unhygienic  
condition...etc");  
deepika.cRequestRefund(dish, restaurant, deepika);  
ApiFood.aRefund(deepika);
```

```
Else  
    restaurant closed;
```

