Modern Method Software Engineering

Home work 3

Shanbo Li	**840810-A478**	**shanboli@Gmail.com**	**0704646157**
Sike Huang	**850414-A394**	**sike.huang@gmail.com**	**0762320173**

2007-9-30

Question 1: Draw class diagram corresponding to this sequence diagram

Answer：

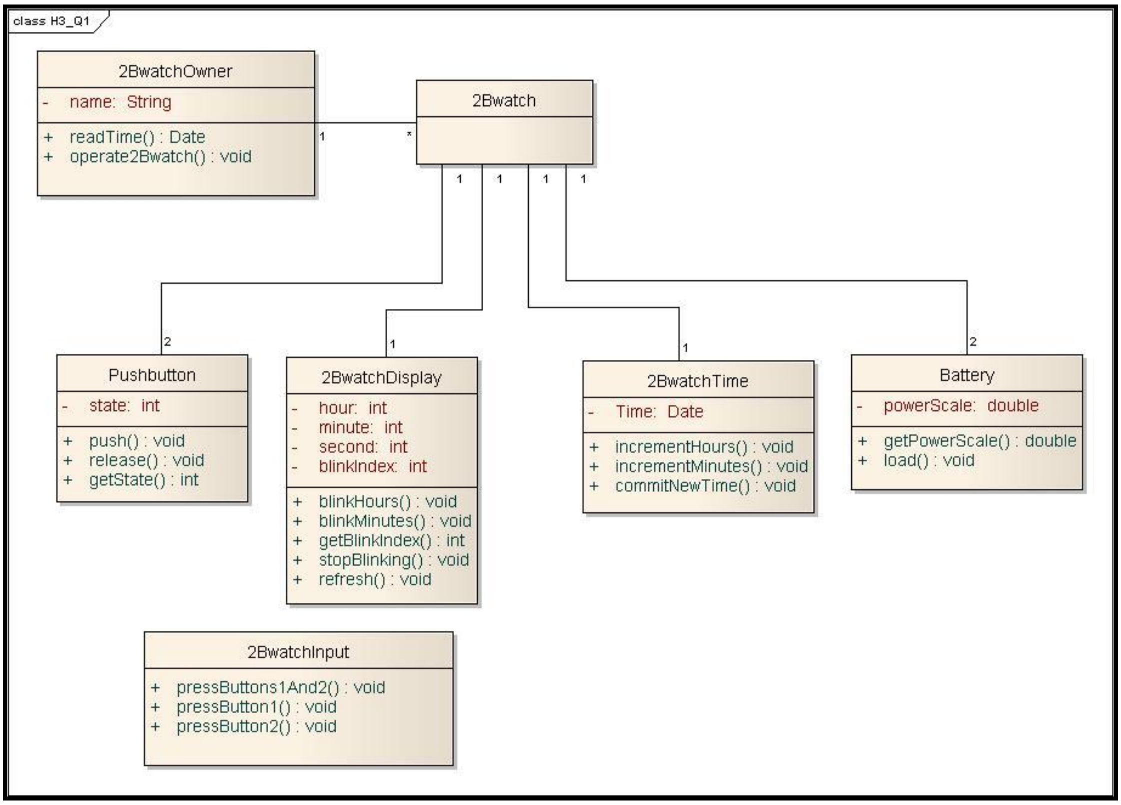

Figure 1-1 2Bwatch class diagram

The sequence diagram is not a good sequence diagram, for it is not clear which class is boundary and which is control. As this question require draw a class diagram corresponding to this sequence diagram. We draw the class diagram *Figure 1*. And we also draw a class named 2BwatchInput as a control like class(if it is a real control class, it should be at the 3rd column of sequence diagram).

Question 2: Consider "Academic Payroll System for University" (Introduced in Question 4 of Homework 2) and based on example on page 206 of textbook, develop a requirement analysis document which includes:

1. Identified objects (entity, boundary, control) also describe the heuristics you use.

Answer:

Identify Entity Objects

We identify entity objects and their attributes by applying Abbott's heuristics on the problem statement and the use cases defined in last homework. We focus on recurring noun phases that denotes concepts of application domain, and keep eyes on real world entities as well.

Entity Object	Attributes & Associations	Definition
AcadamicPayroll	name of university	An instance of Academic Payroll System.
Academic	name (implied)	An actor presenting teaching staff or lab researcher.
FullTimeAcadamic		An Academic who works full time.
CasualAcadamic		An Academic who works part-time according to Contract.
Employee	name (implied)	An actor presenting administrative staff under personnel department.
Leave	balance	A Leave stands for the facts that a staff is off work.
Payment	tax rate method	A Payment is a transaction of money to staff.
Salary	amount (implied)	A Salary is the amount of money to be paid to staff.
Contract	number hourly rate	An agreement signed by CasualAcadamic.

Table 2-1 Entity Objects of AP system

Identify Boundary Objects

We scan the use cases and problem statement and identify the different points where information is exchanged between the actors and the system. We focus on interfaces and notices in which the system provides information to the actors.

Boundary Object	Definition
Desktop	**Windows-based desktop interface used by Employee to access AP System.**
AcadamicKiock	**Web-based interface used by Academic to access AP System.**
PaymentNotification	**Notification received by CasualAcadamic about the Payment.**
TimeCard	**CausualAcdemic uses TimeCard to notify AcadamicPayroll about dates and hours worked.**

Table 2-2 Boundary Objects of AP system

Identify Control Objects

Control objects represent the coordination among boundary and entity objects, we consider two use cases in this problem. In ViewPaymentDetail, we identify a single control object called **ViewPaymentControl**, which is responsible for collecting all history payments, and checking individual payment. In MakePayment, we identify a single control object called **MakePaymentControl**, which is responsible for figuring out salary, generating payment, and finally notifying user.

2. UML sequence diagrams for the problem.

Answer:

Notations used in sequence diagram:

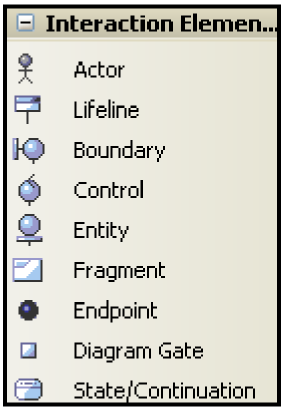

Figure 2-1 Notations

UML sequence diagram for ViewPaymentDetail use case

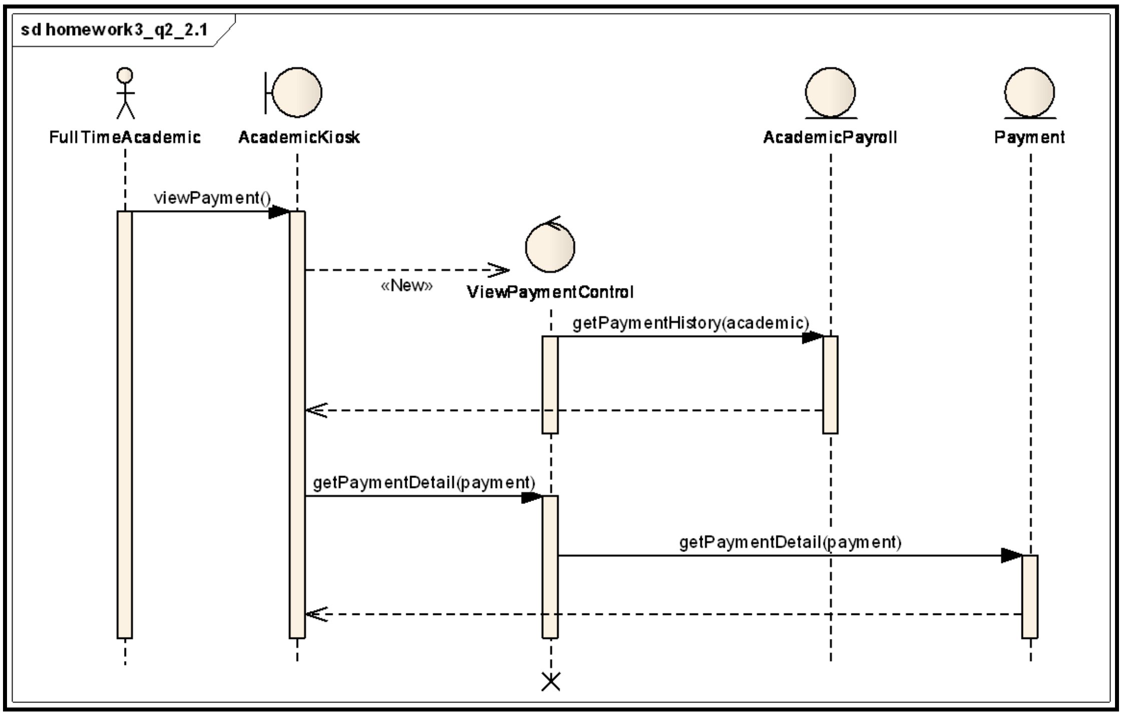

Figure 2-2 sequence diagram for ViewPaymentDetail use case

When requesting payment details, the control object must first obtain a list of (history) payments. It requests from the AcademicPayroll class, which maintains the list of (history) payments. Then for each payment, the control object consults the Payment class for details.

UML sequence diagram for MakePayment use case

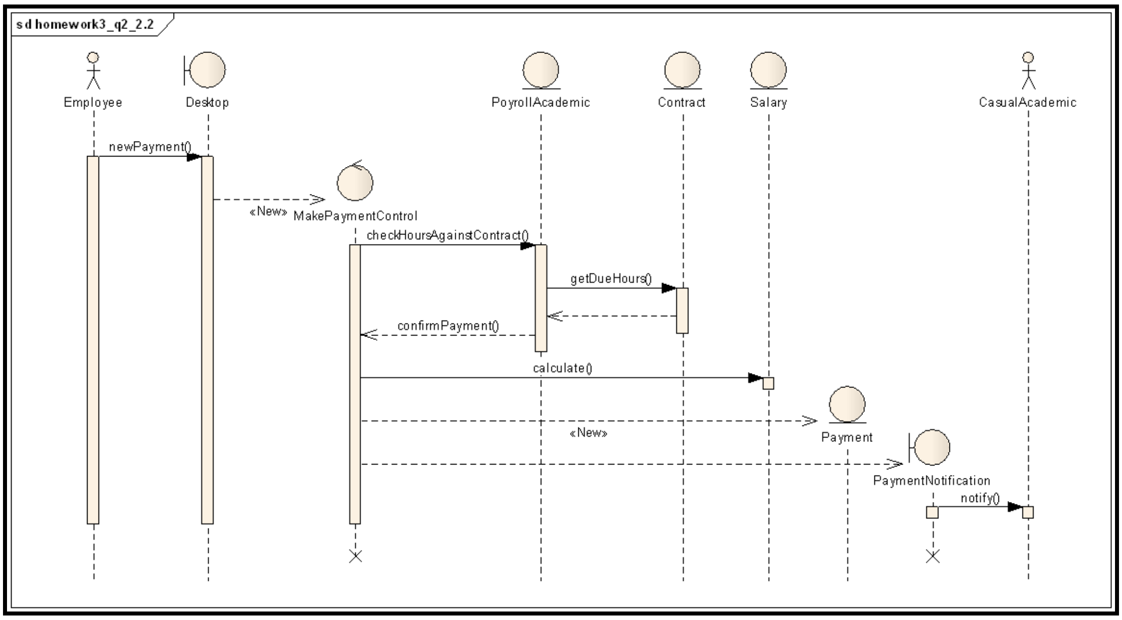

Figure 2-3 sequence diagram for MakePayment use case

When making new payment, the control object first verifies working hours in Contract via PayrollAcademic, and calculates the corresponding salary, then creates a new instance of Payment entity, finally it produces a new boundary object PaymentNotification to notify CasualAcdemic about the arrival of new payment.

3. UML class diagrams for the problem.

Answer:

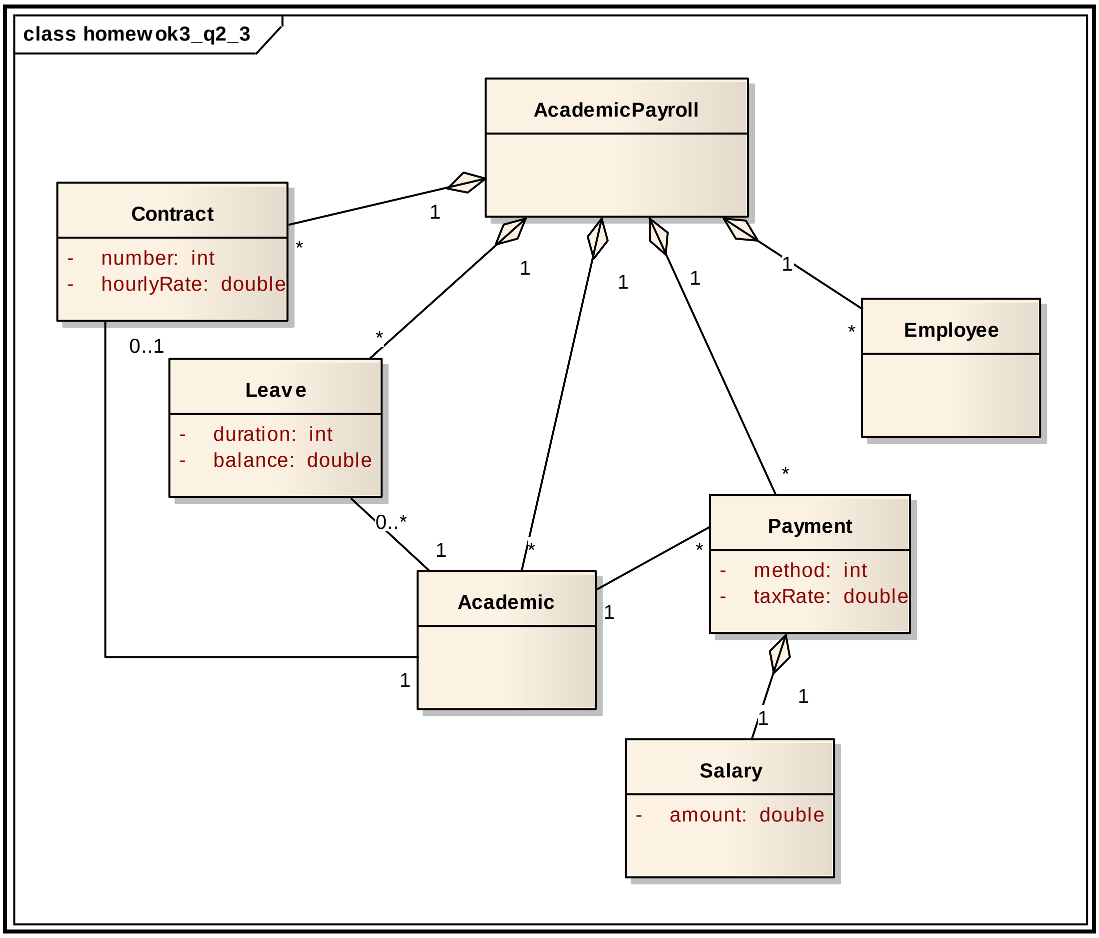

Figure 2-4 class diagrams for the problem

In this UML class diagram, we focus on associations only.

AcademicPayroll tracks a list of all Conrtacts, Leaves, Academics, Payments and Employees. An Academic has no Leave (full-time Academic) or more than one Leaves (part-time Academic), only part-time Academic has a Contract, and An Academic has many Payments, where each Payment contains a piece of Salary.

4. Inheritance hierarchy among entity objects.

Answer:

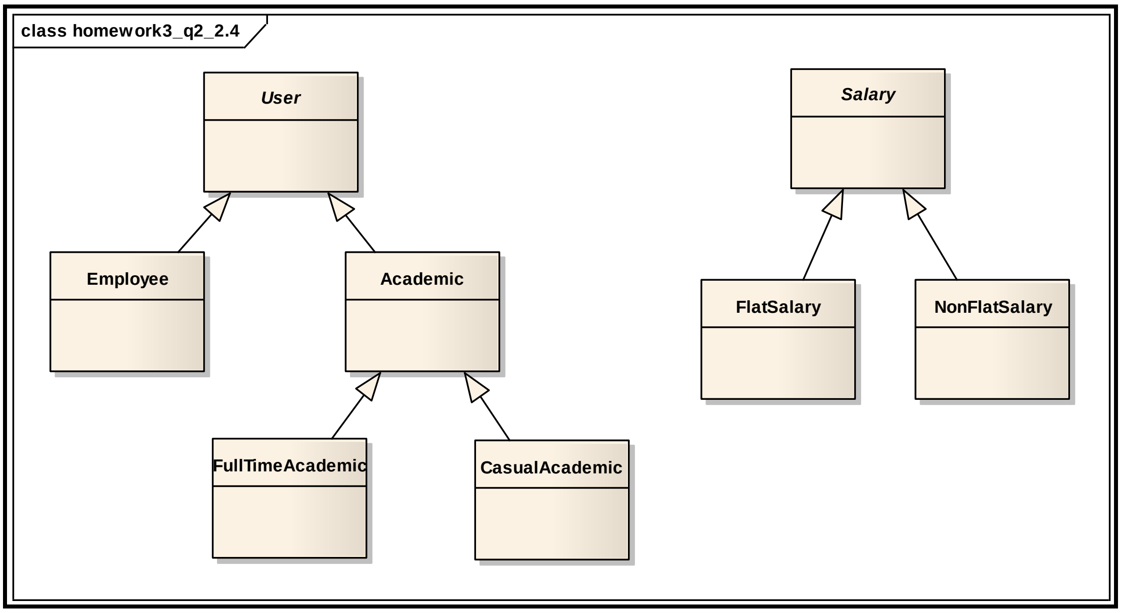

Figure 2-5 Inheritance hierarchy among entity objects

First, we identify an abstract class User through generalization, Employee and Academic are two types of User who access AcademicPayroll system differently, FullTimeAcademic and CasualAcademic are kinds of Academic having distinct Salary, FlatSalary for FullTimeAcademic while NonFlatSalary for CasualAcademic.

Question 3: Draw state-chart describing the behavior of the AnnounceTournamentControl object based on the sequence diagrams of figure 5-26 through 5-28. (Bonus point 01).

Answer:

We rearranged some state, for the first sequence of Figure 5-28 (from notifySponsorsOfDecision to notifyAdvertiser) should be in the FindSponserShip substate.

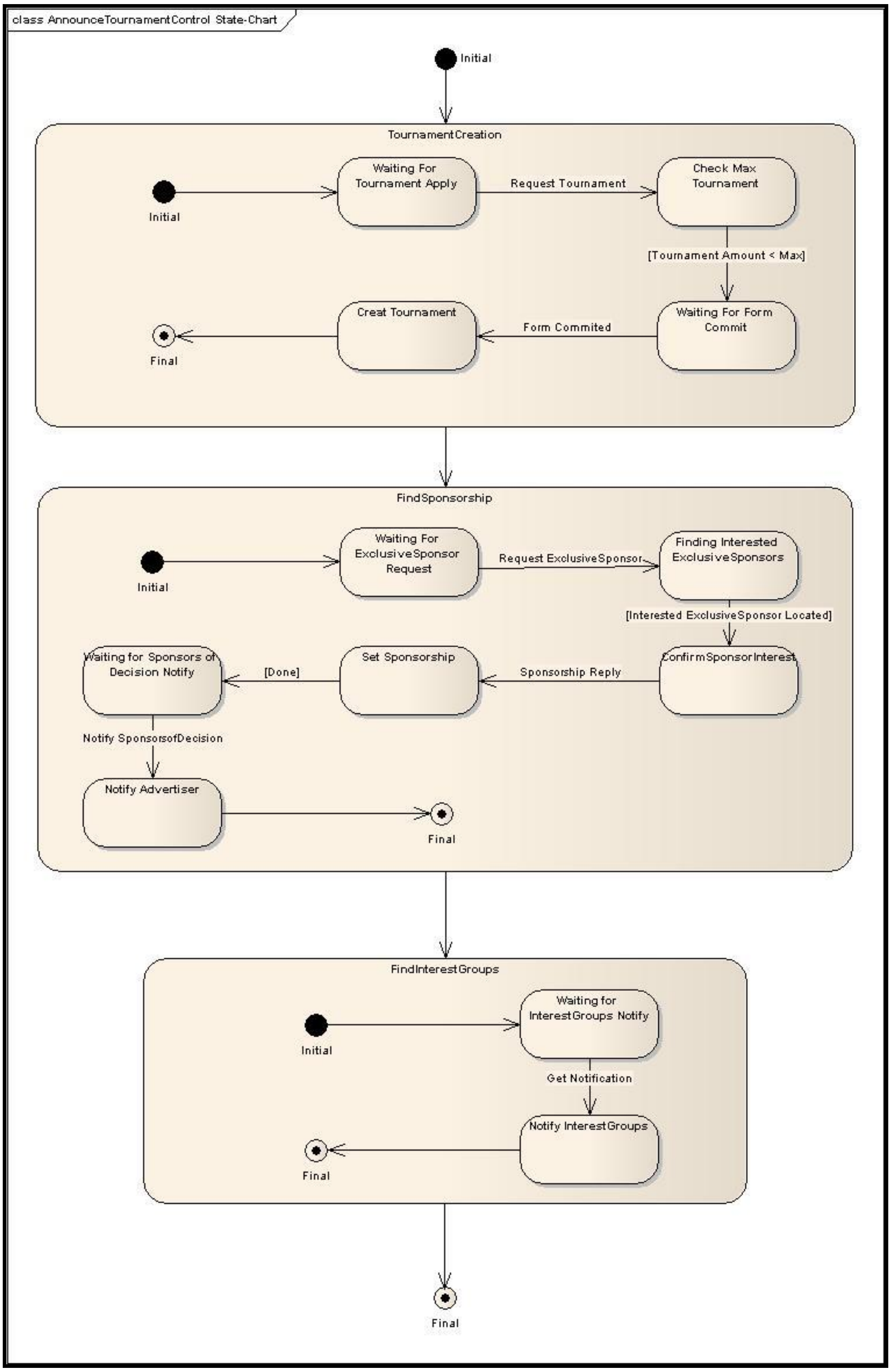

Figure 3-1 state-chart diagram of AnnounceTournamentControl

Question 4:

1. Identified entity objects.

Answer：

Entity Object	Attributes & Associations	Definition
AdvertConsult	name of advertising consultancy company	An instance of Advert Consultancy System.
Campaign	start date end date cost status associated client	A Campaign is a competition among advertising companies.
Advertisement	type	Advertisements build up a Campaign.
Record	id associated campaign	Records are kept for all Clients.
Client	name (implied)	An Actor to which Advert Consultancy Inc. supports in advertising campaigns.
Director	name of department	An Actor in response of negotiating with Client and giving Assignment to Staff.
Salary	amount	
Staff	pay grade annual bonus skill set	An Actor in charge of consultancy to Client.
Assignment	associated tasks	An Assignment is given by Director to Staff.
Task		A work to be performed by Staff.

Table 4-1 Entity Objects

2. Identified boundary objects.

Answer:

Boundary Object	Definition
NegotiationTemplate	Form used by Director and Client to make a Campaign.
RequestNote	Form used by Director to assign Staff among departments.
Acknowledgement	Notification received by Director about the assignment of Staff.
TimesheetForm	Staff uses TimesheetForm to notify AdvertConsult about hours of consultancy performed.

Table 4-2 Boundary Objects

3. Identified control objects.

Answer:

In AssignTask, we identify a single control object called **AssignTaskControl**, which is responsible for forwarding task assignment to another director if necessary and creating new assignment. In UpdateTimesheet, we identify a single control object called **UpdateControl**, which is responsible for classifying in-office and out-of-office hours and updating the timesheet correspondingly. In OrganizeCampaign, we identify a single control object called **OrganizeCampaignControl**, which is responsible for checking previous records of campaigns and creating new campaign with associated resources.

4. Map the use cases you developed in Homework 2 to identified objects with sequence diagrams.

Answer:

UML sequence diagram for AssignTask use case

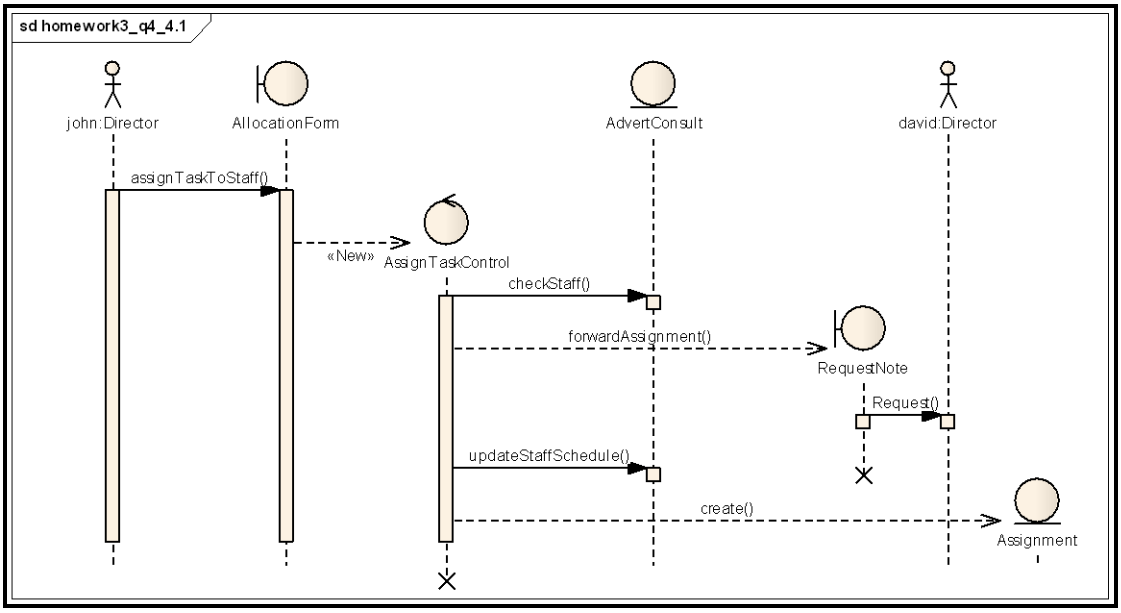

Figure 4-1 sequence diagram for AssignTask use case

Notice we simplify the sequence diagram a little bit in this use case. Two Director john and david from different apartments are involved in AssignTask use case, john interacts with the system with AllocationForm, a control object AssignTaskControl is generated to check the Staff on entity object AdvertConsult, then it forwards the request to another Director by creating a new boundary object RequestNote, and finally makes a new Assignment entity.

UML sequence diagram for UpdateTimesheet use case

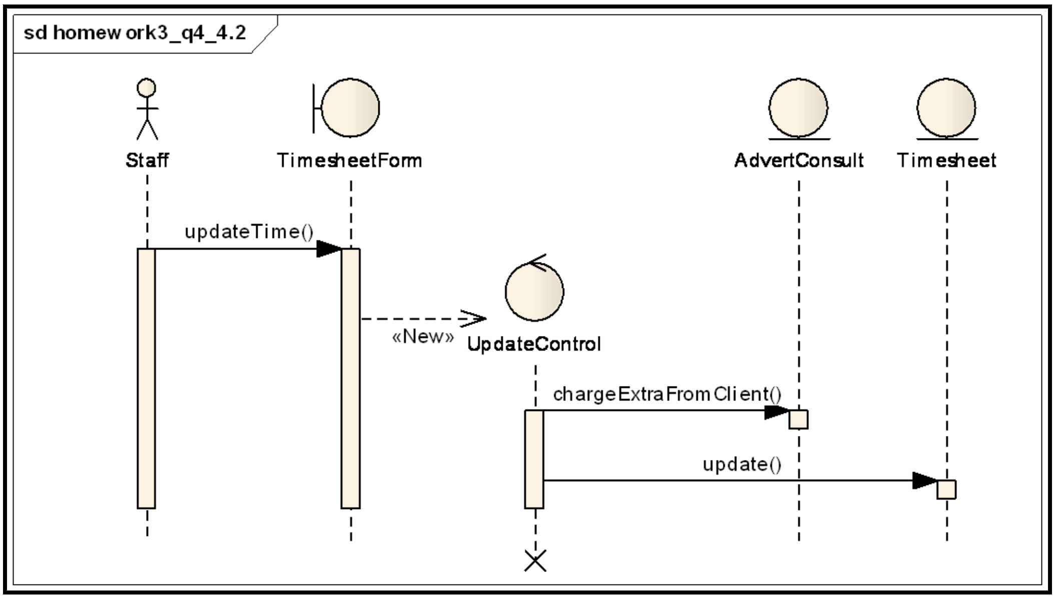

Figure 4-2 sequence diagram for UpdateTimesheet use case

When updating hours worked, the control object first check the hours of out-of-office work done by Staff, and asks entity object AdvertConsult to charge extra from Client, and update the Timesheet entity by in-office and out-of-office hours in the end.

UML sequence diagram for OrganizeCampaign use case

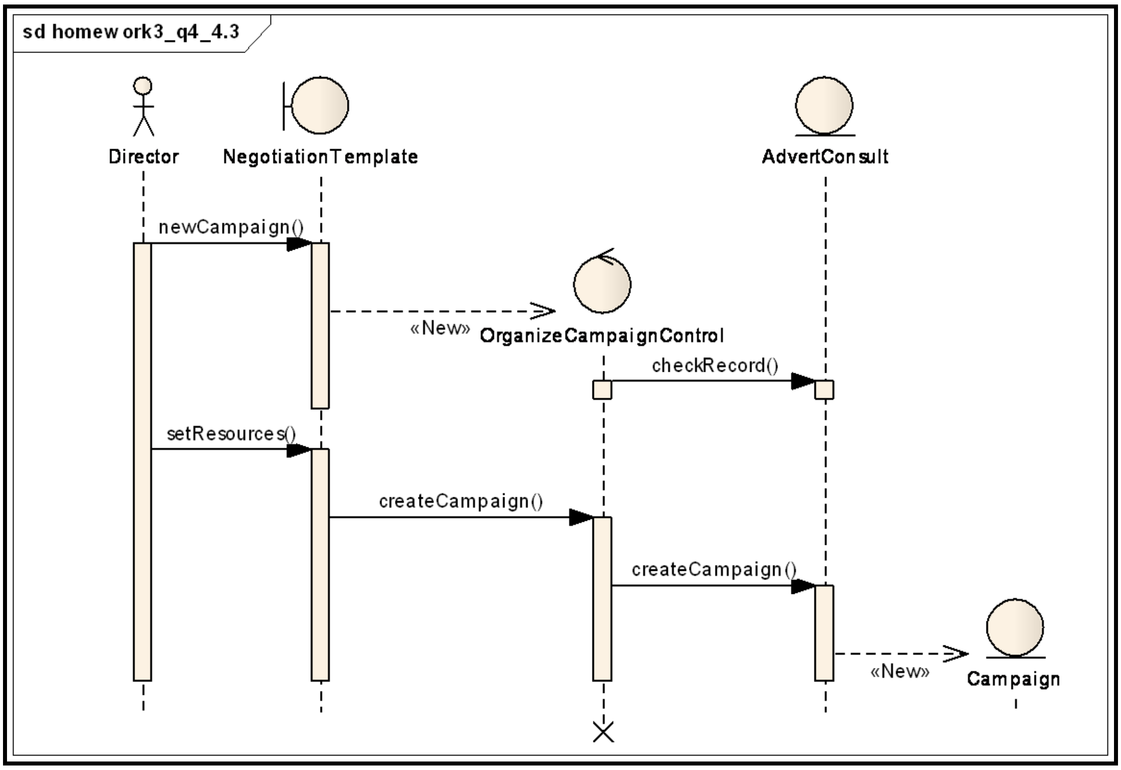

Figure 4-3 sequence diagram for OrganizeCampaign use case

Notice we simplify the sequence is this diagram, when creating an advertising campaign, the control object check the records the previous campaign from entity AdvertConsult, after Direct specifies all the resources to the campaign, the control object tells AdvertConsult to create a new entity Campaign.

5. UML class diagrams.

Answer:

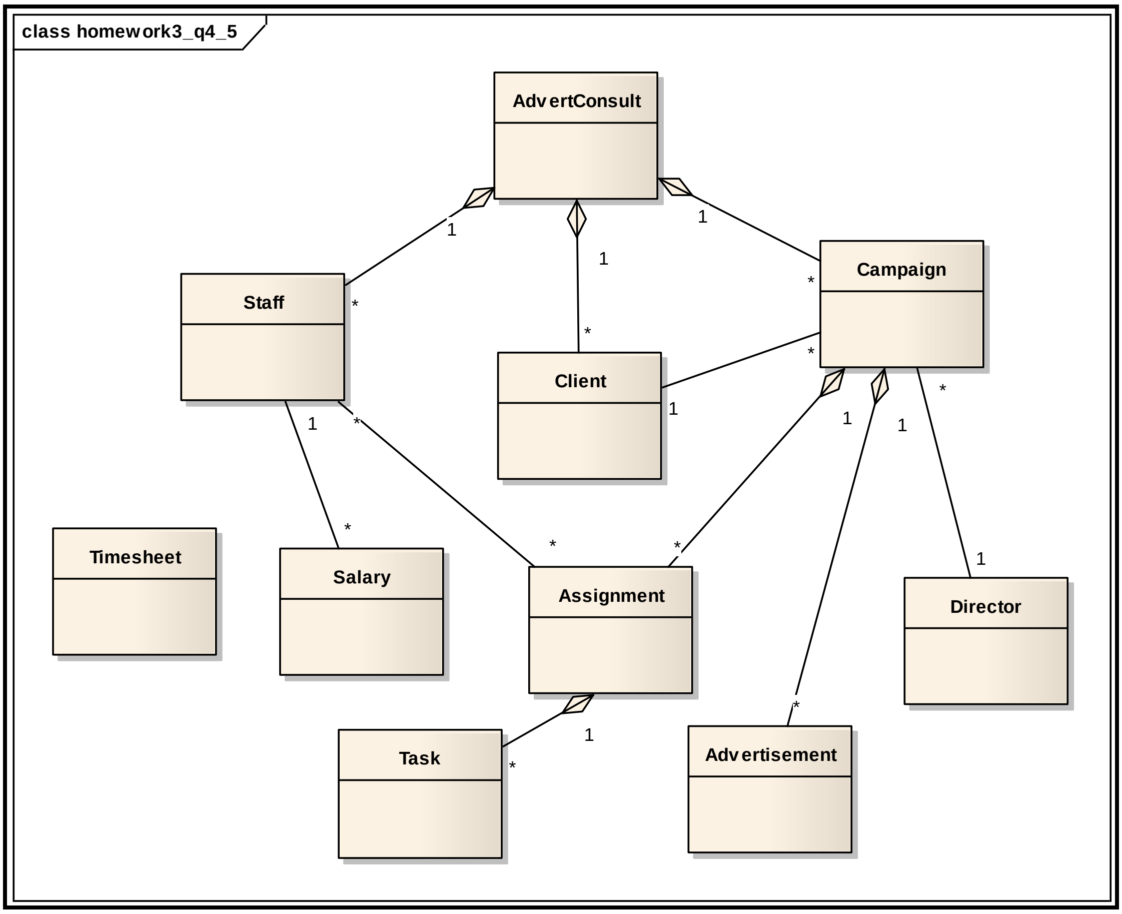

Figure 4-4 class diagrams

In this UML class diagram, we focus on associations only.

From AdvertConsult, we can obtain the list of Clients, lists of advertising Campaigns with respective Client, and lists of Staffs with their Salaries. Each Campaign is conducted by one Director, composing of several Advertisements, and many Assignments are put into one Campaign with a set of Tasks and Staffs in charge.

6. Inheritance relationships among objects identified in step 1-3.

Answer:

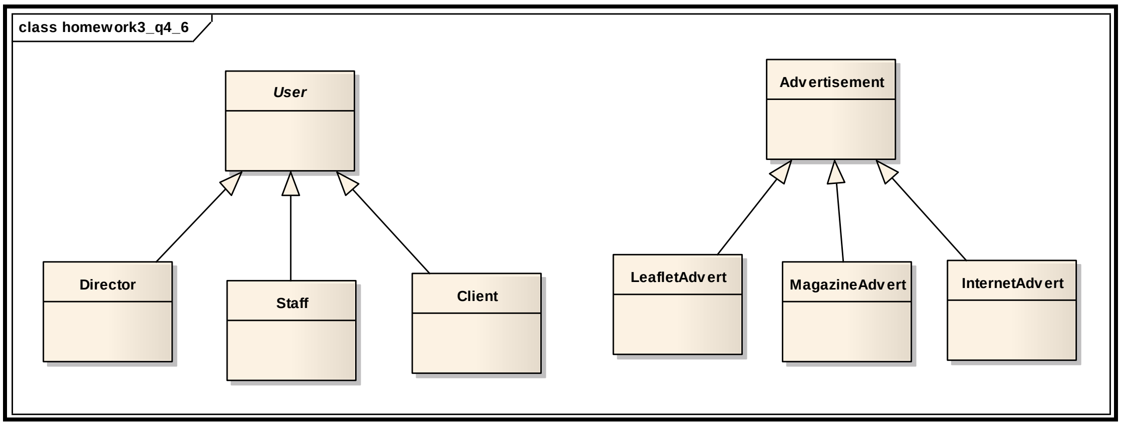

Figure 4-5 Inheritance relationships

First, we identify an abstract class User through generalization, Director, Staff and Client are three types of User who access AdvertConsult system differently, and this enables us to treat common attributes of various users in a more general fashion. We identify another inheritance hierarchy, Advertisement, identified through specialization to provide examples for concepts, the LeafletAdvert, MagazineAdvert and InternetAdvert are concrete specializations of Advertisement that using different media.

7. State-chart diagrams for objects identified in step 1-3.

Answer:

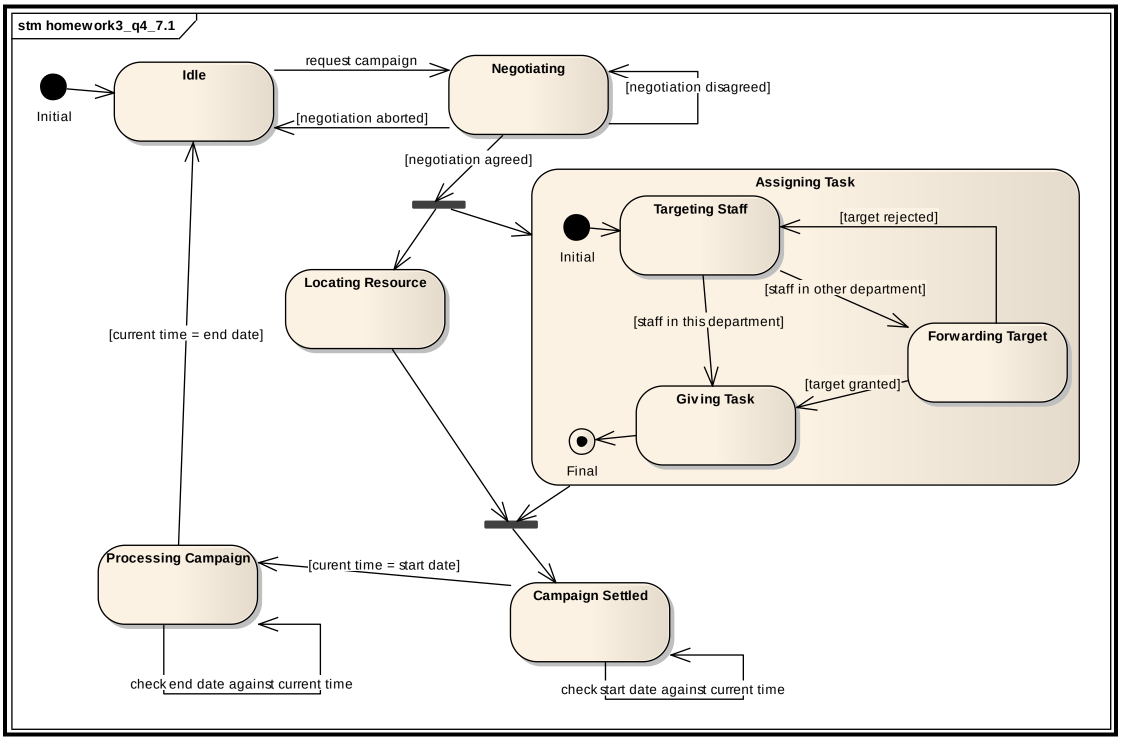

Figure 4-6 State-chart diagram of AdvertConsult

The diagram above shows the states that **AdvertConsult** may traverse, once the negotiating state is passed, the states locating resource and assigning task are executed in parallel, while in state assigning task, a flow of sub states are entered to ensure tasks are given to corresponding staffs, finally, the campaign will be launched once start time arrives and terminated when end time occurs.

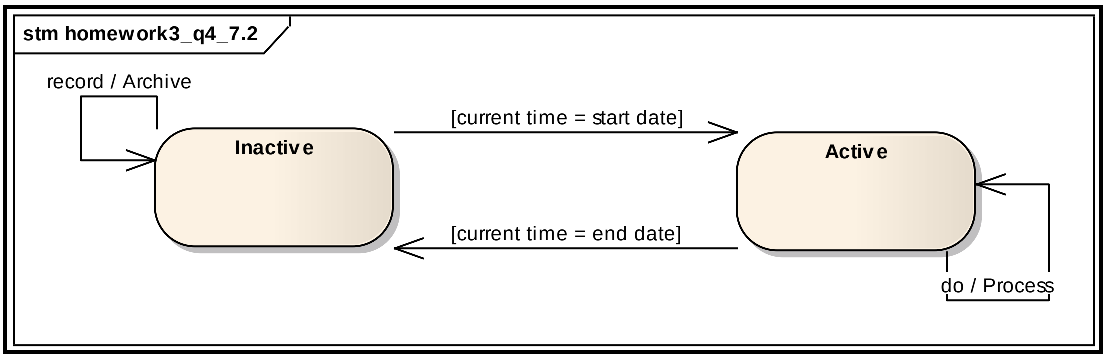

Figure 4-7 two states owned by Campaign

The diagram above illustrates two states owned by **Campaign**, a campaign is either active or inactive, in accordance with the internal system time, when active, campaign is processing, whilst in inactive state, previous campaign will be archived as history.