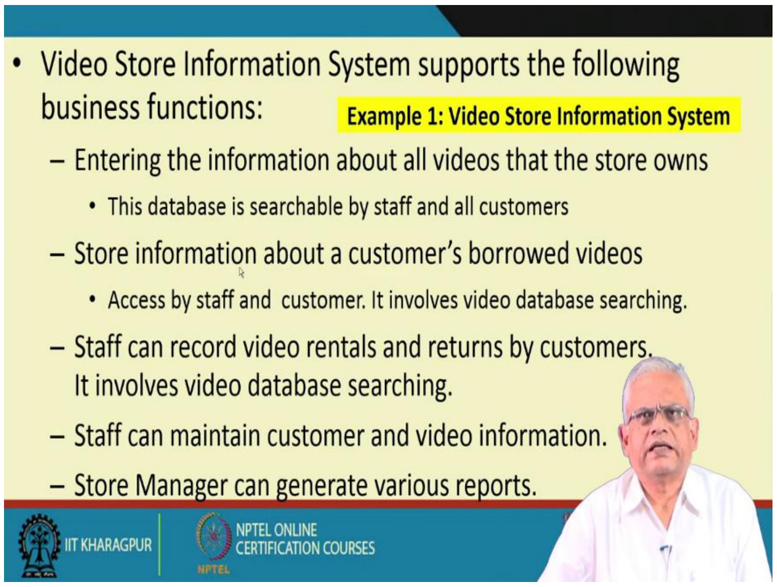
Object-Oriented System Development Using UML, JAVA and Patterns

Professor Rajib Mall
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Use Case Examples
Lecture 05

Welcome to this lecture. In the last lecture we had discussed some of the essential concepts of Use Case modelling and we had looked at the basic diagrams on how to factor a Use case and so on. And at the end of the lecture, we were discussing about an example problem. The problem was about a video store information system.

(Refer Slide Time: 0:43)



- Video Store Information System supports the following business functions:

Example 1: Video Store Information System

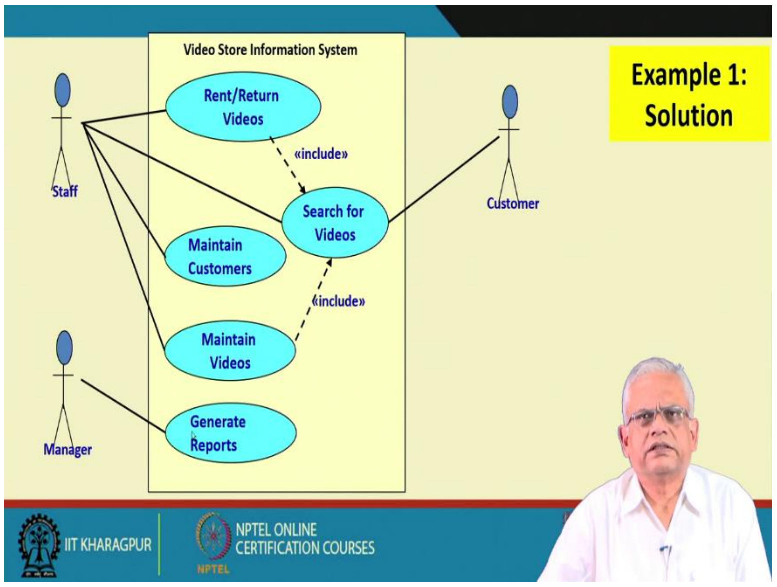
  - Entering the information about all videos that the store owns
    - This database is searchable by staff and all customers
  - Store information about a customer's borrowed videos
    - Access by staff and customer. It involves video database searching.
  - Staff can record video rentals and returns by customers. It involves video database searching.
  - Staff can maintain customer and video information.
  - Store Manager can generate various reports.

In the video store information system, we had a text description about the stored operations and we are required to find out from the text description about what are the Use cases and the actors and then represent them in the form of a Use case. The first functionality here of course in this text description, we have almost clearly identified the different functionalities in the form of bullet points. But then in practice you may get a text description in the form of a paragraph or a page. And then have to read it thoroughly and identify the use cases and remember that each use case is a functionality of the system that can be invoked by a user. So, let us read this.

The first functionality is about entering information about all the videos that the store owns. And once the video information is stored, the database can be searched by staff and customer. That means if a customer or a staff wants to find out if a video exist in the store, he can search the database. The second is stored information about the borrowed videos. So, this is what are the videos borrowed by a specific customer. This is accessed by staff and customer involves video database searching. The staff can record video rentals and returns by customers. So, each time a customer borrows a video, this is entered by the staff and also it is written by the customer, this information is entered by the staff.

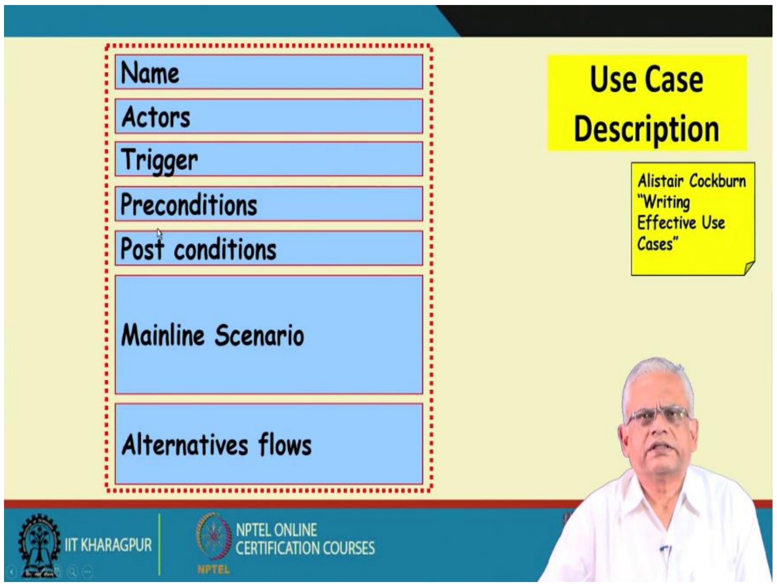
So, we have both the actor that is staff and customer, and the use case that is record video borrowing or video rental. The staff can maintain customer and video information that is can add new customers, delete customers, similarly can add new videos, delete some of the videos, that is the maintenance activity. And the store manager can generate various reports. If we identify these use cases and the actors, we should be able to represent in the form of a Use case diagram. And of course, we need to identify any factoring of the use case. Let us represent this in the form of a Use case diagram for this example.

(Refer Slide Time: 4:20)



We have the rent/return videos is by the staff as mentioned and the rent and return use case includes the search for videos. The search for videos is also can be invoked by the customer and the staff. The staff can maintain the customers, that is add customer, delete customer and the staff can maintain the videos, that is add videos, delete videos and this involves searching the video database. And the manager can generate reports. So, this captures all the functionality has been defined in the text description.

(Refer Slide Time: 5:22)



Use Case Description

Alistair Cockburn "Writing Effective Use Cases"

Name

Actors

Trigger

Preconditions

Post conditions

Mainline Scenario

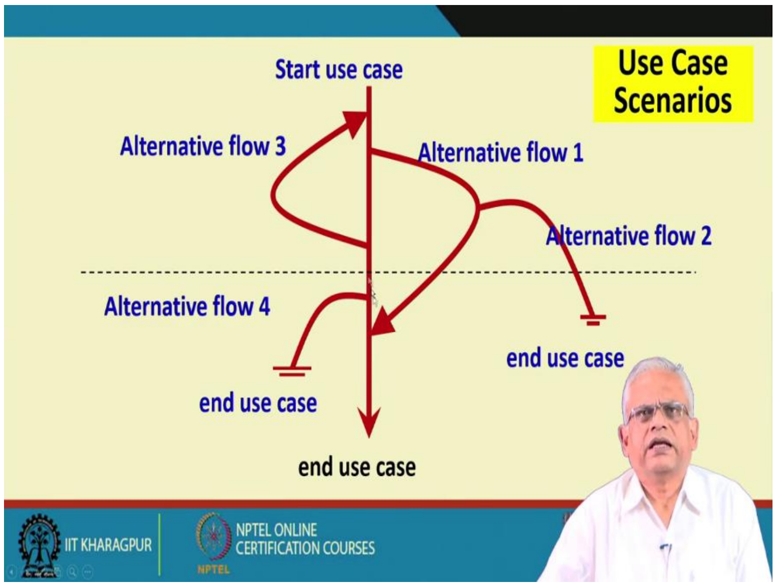
Alternatives flows

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

But the use case model or the use case diagram gives only limited amount of information. More specific information needs to be accompanied in the form of a text description. The UML standard by itself does not recommend any specific style for recording the information to accompany the Use case diagram. But there are some standard ways have emerged based on work of some authors and companies. One is about Alistair Cockburn "Writing Effective Use Cases" book. So, in this book Alistair Cockburn recommends that the description accompanying a use case diagram should have name of the use case, the actors who would be participating in the use case execution.

The triggers that is under what instance or events the use case will start, the precondition for the use case to start i.e., what condition should have been satisfied. The post condition after the use case completes, what are some of the conditions that should be satisfied and the different scenarios in the use case. The main line scenario is the typical scenario but then the exceptional scenarios are called as the alternative flows and these also should be identified for each use case. We will just take couple of examples and see how we can document use case.

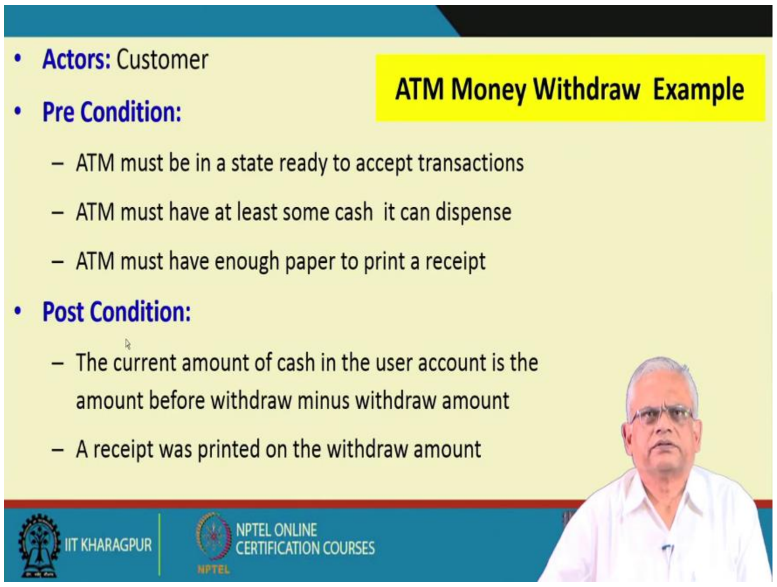
(Refer Slide Time: 7:37)



But before we discuss how to document, we need to discuss about the use case scenarios. Almost every use case has a number of scenarios, one is called the main line scenario. That we have shown in the form of a straight arrow in the diagram. Typically, this is what happens. You want withdraw cash from the ATM, just went there and then inserted the card and it ask for pin, entered the pin, entered the amount and then confirmed and it dispense the cash.

Typically, that is what happens. But sometimes you enter the ATM card, entered password, it asked for the amount, you entered 550 rupees, it said: please enter amount in multiple of 500. So, you have to re-enter. That is one scenario. And then you enter correctly and it dispense the cash. In another scenario, it may so happen that after you enter the amount, it says: insufficient balance and it ends the use case, just ejected the card and so on. So, for every use case we need to identify the main line scenario that typically occurs and the alternate scenarios.

(Refer Slide Time: 9:30)



- **Actors:** Customer
- **Pre Condition:**
  - ATM must be in a state ready to accept transactions
  - ATM must have at least some cash it can dispense
  - ATM must have enough paper to print a receipt
- **Post Condition:**
  - The current amount of cash in the user account is the amount before withdraw minus withdraw amount
  - A receipt was printed on the withdraw amount

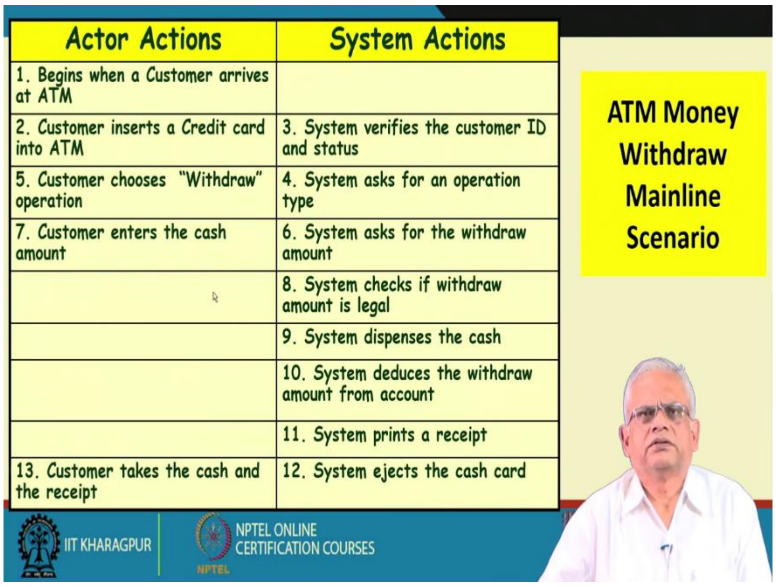
ATM Money Withdraw Example

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL

This is an example documentation, text document to accompany the use case diagram for the ATM. Here, the use case that we are documenting is money withdraw. The actor is the customer. The precondition is the condition that must be satisfied for the use case to operate. Here, the ATM must be in a ready state to accept transaction. ATM must have some cash, ATM must have enough paper to print. Otherwise, the money withdraw use case cannot be started. The post condition is the conditions that holds after the use case completes. Here, the post condition can be the current amount of the cash in the user account and the amount before withdraw cash minus the withdraw amount. So, that is after the cash has been withdrawn. Then that must reflect in the user account. A receipt was printed on the withdraw amount.

But this appears bit artificial both precondition and post condition. This is just an example. And we need to write the precondition and post condition when it adds value to text description. It is not necessary that we just keep on writing something which does not help in the documentation. Somebody reading the precondition and post condition should be able to get some value regarding the understanding of the use case. Just filling precondition and post condition does not help. Here we have just written this to explain what needs to be written but then unless it adds value we need not have the precondition and post condition.

(Refer Slide Time: 11:49)



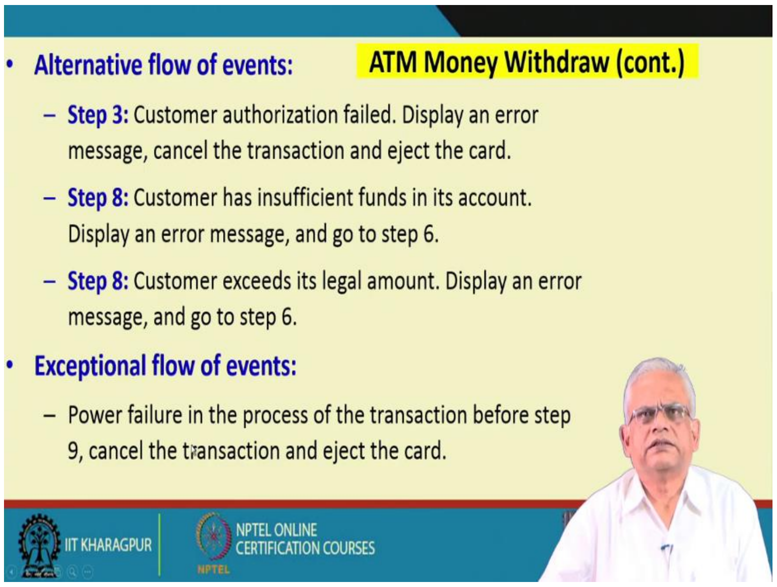
ATM Money Withdraw Mainline Scenario

Actor Actions	System Actions
1. Begins when a Customer arrives at ATM	
2. Customer inserts a Credit card into ATM	3. System verifies the customer ID and status
5. Customer chooses "Withdraw" operation	4. System asks for an operation type
7. Customer enters the cash amount	6. System asks for the withdraw amount
	8. System checks if withdraw amount is legal
	9. System dispenses the cash
	10. System deduces the withdraw amount from account
	11. System prints a receipt
13. Customer takes the cash and the receipt	12. System ejects the cash card

Now, we need to identify the main line scenario and the alternate scenarios. Here, each scenario is in the form of an actor action that is the user comes and input certain thing to the system. And the system responds. So, this is an actor or the user action in response to which the system produces some action. Begins when a customer arrives at ATM, inserts card into the ATM, system verifies the customer id and status, customer chooses the withdraw operation.

The system asks for operation type, the customer enters cash amount. Okay, the system asks for the cash amount that in response to that the customer enters the cash. The system checks if the amount is legal, that is a valid amount. System dispenses the cash. System deducts the withdraw amount from the account and the system prints a receipt. The system ejects the cash card and the customer takes the cash card and the receipt. This is a typical scenario and this is what normally happens in the cash withdraw use case. But there can be alternate scenarios.

(Refer Slide Time: 13:39)

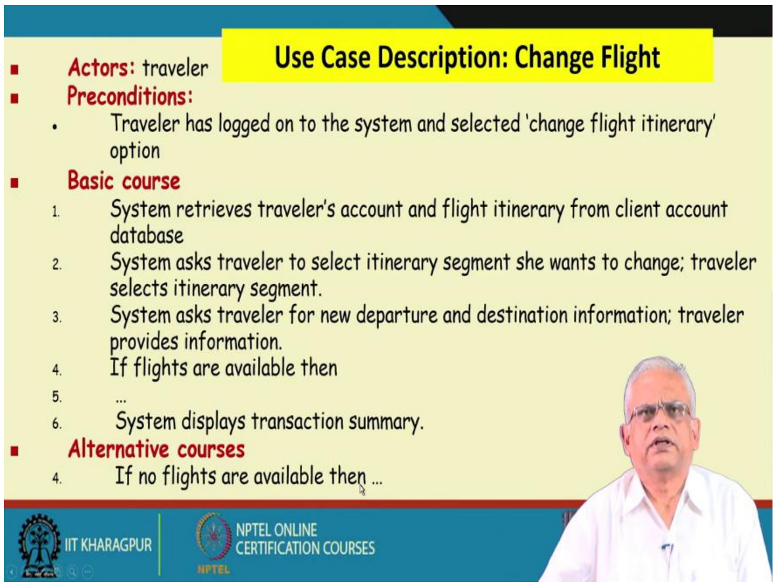


ATM Money Withdraw (cont.)

- Alternative flow of events:
  - Step 3: Customer authorization failed. Display an error message, cancel the transaction and eject the card.
  - Step 8: Customer has insufficient funds in its account. Display an error message, and go to step 6.
  - Step 8: Customer exceeds its legal amount. Display an error message, and go to step 6.
- Exceptional flow of events:
  - Power failure in the process of the transaction before step 9, cancel the transaction and eject the card.

One alternate event is that at step 3, the customer authorization fails and the pin does not match and displays an error message and cancels the transaction and ejects the card. Another alternate flow is at step 8, the customer has insufficient fund in the account. It displays an error message and it starts from step 6. In step 8, the customer exceeds the legal amount, it displays an error message and goes to step 6. So, the customer enters more than 20000 or something. There can be some exceptional flows of event for example, power failure in the process of transaction, then cancel the transaction and eject the card.

(Refer Slide Time: 14:40)



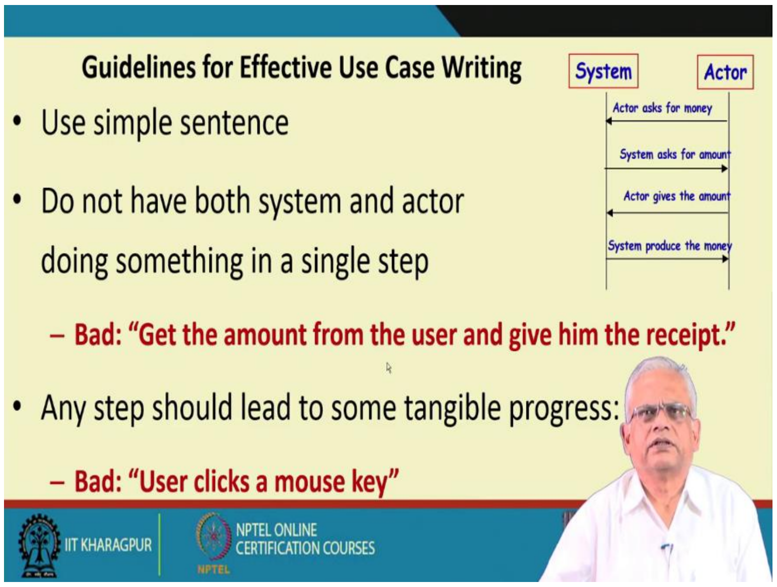
Use Case Description: Change Flight

- Actors: traveler
- Preconditions:
  - Traveler has logged on to the system and selected 'change flight itinerary' option
- Basic course
  1. System retrieves traveler's account and flight itinerary from client account database
  2. System asks traveler to select itinerary segment she wants to change; traveler selects itinerary segment.
  3. System asks traveler for new departure and destination information; traveler provides information.
  4. If flights are available then
  5. ...
  6. System displays transaction summary.
- Alternative courses
  4. If no flights are available then ...

This is another example of the description of the change flight in a travel portal. The actor is the traveler, the precondition is that the traveler has logged into the system and selected flight change, change flight itinerary. The main line course or the basic course here is that the system retrieves the account, travelers account, the flight itinerary and then it asks the traveler to select the segment he want to change. The system asks the traveler new departure and destination information and if flight is available, then it makes the necessary changes if additional cost to be incurred asks for the cost.

Otherwise, if there is no alternate flight available it just stops and displays the transactions summary. So, if no flight is available, then it is an alternate course. If the flights are available, then it proceeds for booking with the amount payable and so on and displays the transaction summary. But it may so happen that there is no flight available and then that is an alternate course and it just displays that no flight available and it stops.

(Refer Slide Time: 16:16)



Guidelines for Effective Use Case Writing

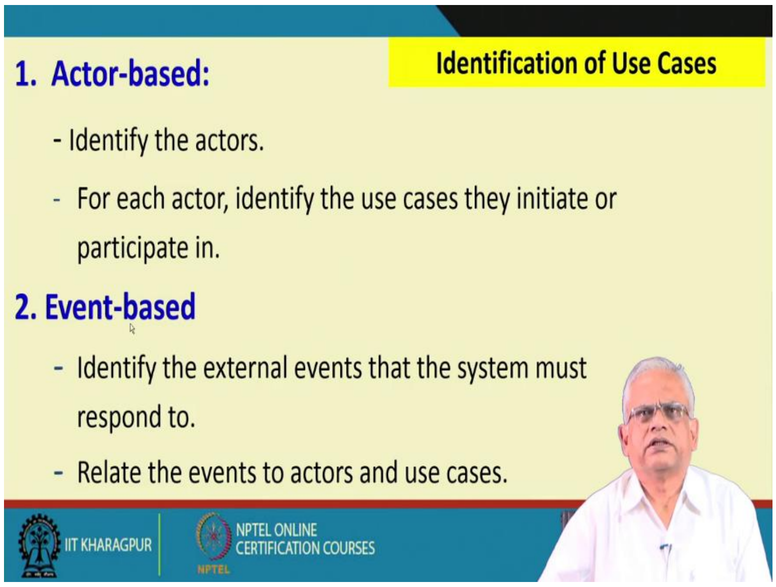
- Use simple sentence
- Do not have both system and actor doing something in a single step
  - Bad: "Get the amount from the user and give him the receipt."
- Any step should lead to some tangible progress:
  - Bad: "User clicks a mouse key"

System
Actor
Actor asks for money
System asks for amount
Actor gives the amount
System produce the money
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

In writing the text description, there are some guidelines for effective use case writing which have emerged. Most of the authors, they agree on this that the documentation should use simple sentences. One sentence should not have both the system doing something and the actor responding or the actor and the system responses are not written on a single sentence. For example, get the amount from the user and give him the receipt is not a very good way of documenting. We should write one step is that the system prompts about the amount, the user inputs the amount and the system gives him a receipt or some such thing. So, it has to be split between the user action and the system response.

Just like here, this is actually the system responds to the actor actions. The actor acts and based on that the system produces some response and so on. And the description should also be like that and also we should be careful that should not document trivial events. For example, let us say that the user clicks the mouse key to select an option, selects one of the check boxes and then click another mouse key for selecting another check box and so on. That only becomes adds to the length to the document without adding value we have to write a crisp document, where each step of the user and the system leads to some progress towards the use case execution, some tangible progress it should not be trivial.

(Refer Slide Time: 18:48)



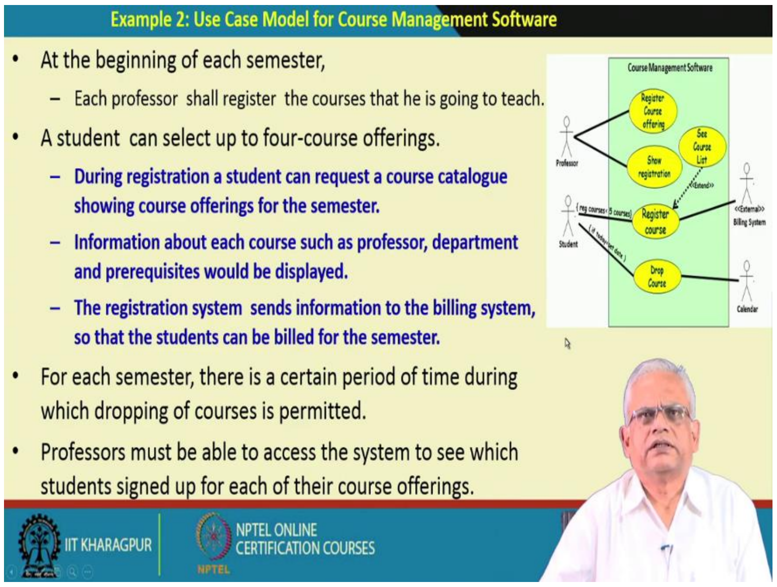
Identification of Use Cases

1. **Actor-based:**
   - Identify the actors.
   - For each actor, identify the use cases they initiate or participate in.
2. **Event-based**
   - Identify the external events that the system must respond to.
   - Relate the events to actors and use cases.

Now the question comes is that given a text description how do we identify that the use cases? One is that we have to read it thoroughly and identify what are the functionality to be supported by the system but then it helps if we can examine the documentation again and actor based way. In the actor based way we first identify who are the users of the system or the actors. And then for each actor or user, we identify which use cases they are initiating and then we document those uses cases in the diagram and also we connect the actor to that use cases. We can also have an event based identification where we identify all the events to which the system responds that is for each use case what can be the input and then based on that we identify the use cases.

And then we identify the events to the actors that which actors generates this event and we connect the actor to the use case. There is no straight forward solution to how to identify the use cases but then the first thing is we have to read the text document thoroughly, try to find out the high level functionalities that the system must support from this reading. And also try out the actor based identification of the use cases and see if we have missed any use case which we previously identified and also try the event based identification and hopefully you would have got all the actors and use cases in the diagram.

(Refer Slide Time: 21:09)



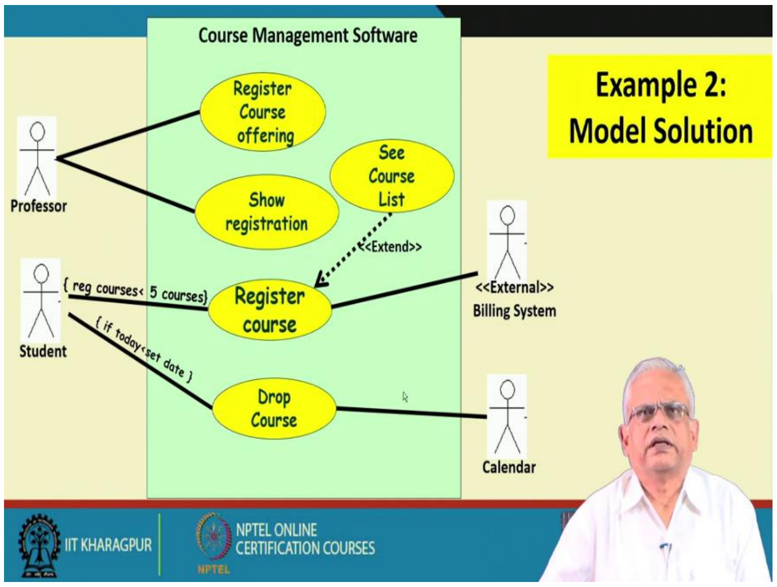
Example 2: Use Case Model for Course Management Software

- At the beginning of each semester,
  - Each professor shall register the courses that he is going to teach.
- A student can select up to four-course offerings.
  - **During registration a student can request a course catalogue showing course offerings for the semester.**
  - **Information about each course such as professor, department and prerequisites would be displayed.**
  - **The registration system sends information to the billing system, so that the students can be billed for the semester.**
- For each semester, there is a certain period of time during which dropping of courses is permitted.
- Professors must be able to access the system to see which students signed up for each of their course offerings.

Now let us try to have another example worked out. This is a course management software. In the course management software, at the beginning of each semester of an academic institute, the professors register for the courses in the software which they are going to teach. So, they sign up for the courses. And once the professor signs up for a course, the students can register for that course. The student can register up to 4 courses and during their registration they can request for a course catalog that is the syllabus which they have to complete and see what are the courses remaining for them. The information about each course such as which professor will offer it, the department, the pre-requisites etcetera. are displayed.

The registration system sends information, once the registration by the student is complete, the software sends information to the billing system which is an external system that is another software which is running so that the student can be billed for the semester. And for each semester there is certain time period during which a student can drop a course he has registered. The professors should be able to find out which students have signed up for their courses. So, this is the rough description of the system. Off course you can have additional description like the student should be the professor should be able to post the grades, a student should be able to see the grades and so on. So, we can document this in this diagram and identify any factoring of the use cases, we will just show it slightly larger in the next slide, so that it is readable.

(Refer Slide Time: 23:41)



So, this is the course management software, the professor registers for the course he is going to offer and the professor can see the courses for which the student have registered. Just observe that all the use cases that we are naming in the verb form, register, show, see, drop, etcetera these are all verbs. The use cases are named in verb form because these are action items. These are the functionality supported by the system.

The two use cases to which the professor can invoke are the register course to be offered and show student registration, who have registered for the courses. And the student and register for course and as the description says, while registering for the course, you can see the course list. So, write the extend this use case, see course list can be optionally invoked during register course. And after course registration, the information is send to an external billing software. So, this is external billing software to which the information is send. And to document that the student can register up to 4 courses, we just write the same thing here. And the student can drop a course till certain time and that we document here in the form of a calendar.

The time even though it appears like the clock is internal to the system, but it is a good practice to have a calendar or clock as one of the actors. And after some time, it is not possible to drop the course. So, whenever there is a time element by certain date, or after a certain date, at certain time and so on, it is good idea to have one of the actor as the calendar or the clock. We will see later that this helps in the design process, the subsequent diagrams we do would be simplified if we have this calendar or clock included here.

And then after doing this, this is the use case diagram which you have developed based on the text description. And at this point, we need to start writing the text description. The text description should accompany this diagram and there we need to use the template that we discussed that for each use case, we write the name of the use case, the actor and if necessary the pre and post conditions and also the main line scenario and all the alternate scenarios must be documented.

Till now we have seen some basic elements of the use case modeling. What are the different symbols in the use case diagram, factoring of the use case and how to identify the use case, we have discussed it with couples of examples. We will just discuss few more examples in the next lecture and identify some of the good principles in the development of use case model and then we will move on to the class diagram. Thank you.