OO Analysis: Use Case Model
Creating Use Case Diagram

A simple case study of an information system of functionality of a student office in accordance the student office of a university with the following specification:

- Many important administrative activities of a university are processed by the student office. Students can register for studies, enroll, and withdraw from studies here.
- Students receive their certificates from the student office. The certificates are printed out by an employee. Lecturers send grading information to the student office. The notification system then informs the students automatically that a certificate has been issued.
- There is a differentiation between two types of employees in the student office:
    - those that are exclusively occupied with the administration of student data (service employee, or ServEmp)
    - those that fulfill the remaining tasks (administration employee, or AdminEmp)

  whereas all employees (ServEmp and AdminEmp) can issue information.
- Administration employees issue certificates when the students come to collect them. Administration employees also create courses. When creating courses, they can reserve lecture halls.

To create a use case diagram from this simplified specification, we perform following activities:

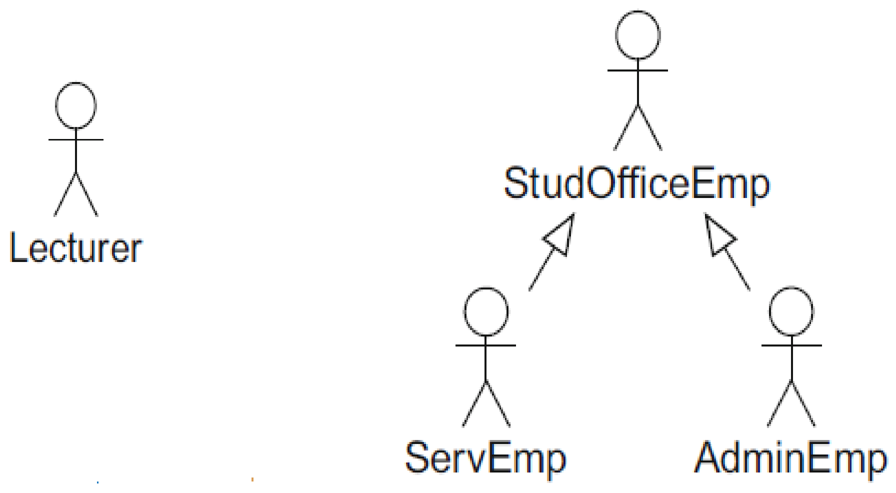
- First identify the actors and their relationships to one another.
- Then determine the use cases and their relationships to one another
- Finally, associate the actors with their use cases

Our objective is to create Use-Case Model of the information system supporting the employees of a student office rather than modeling the functionalities the student office provides for the students.

Identify the Actors (Relationships to one another)
By looking at the textual specification, we can identify five potential actors:

Lecturer
Student
Notification System
employees of the types ServEmp and AdminEmp,

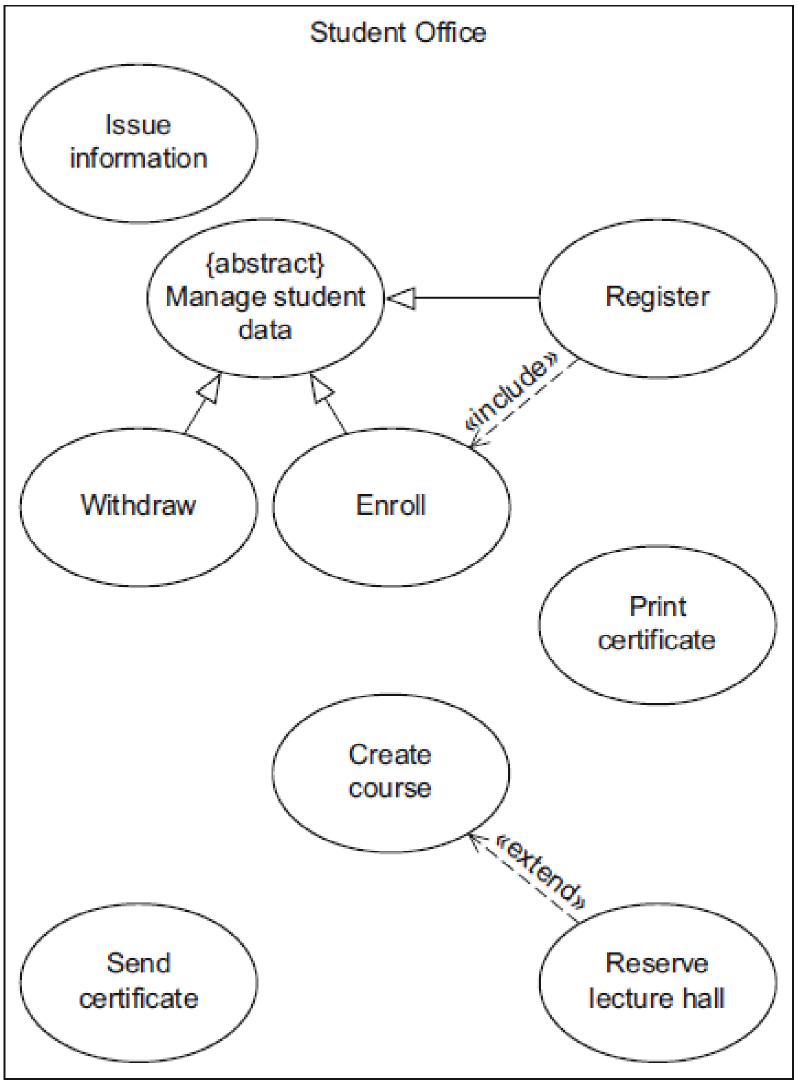
As both types of employees demonstrate common behavior, namely issuing information, it makes sense to introduce a common super-actor StudOfficeEmp from which ServEmp and AdminEmp inherit.
We assume that the Notification System is not part of the student office, rather notification is sent by lecturer action "send certificate" so we do not consider it as an actor (outside the system boundary).
Since we are modeling the information system supporting the employees of a student office, so we will not consider student as an actor to collect certificate.
Final list of actors and their relationships are illustrated below:



Identify Use Cases (relationships to one another)
To identify use cases, we determine which functionalities the student office must fulfill. From the specification, we have the functions Register, Enroll, and Withdraw. We could group these in one use case Manage student data as they are all performed by an actor ServEmp.
Since certificate information (in the form grade) is sent to student office by the lecturer, where a certificate is printed, so we have Print certificate and Send certificate use cases. From system specification we can identify Issue information, Reserve lecture hall, and Create course use cases.

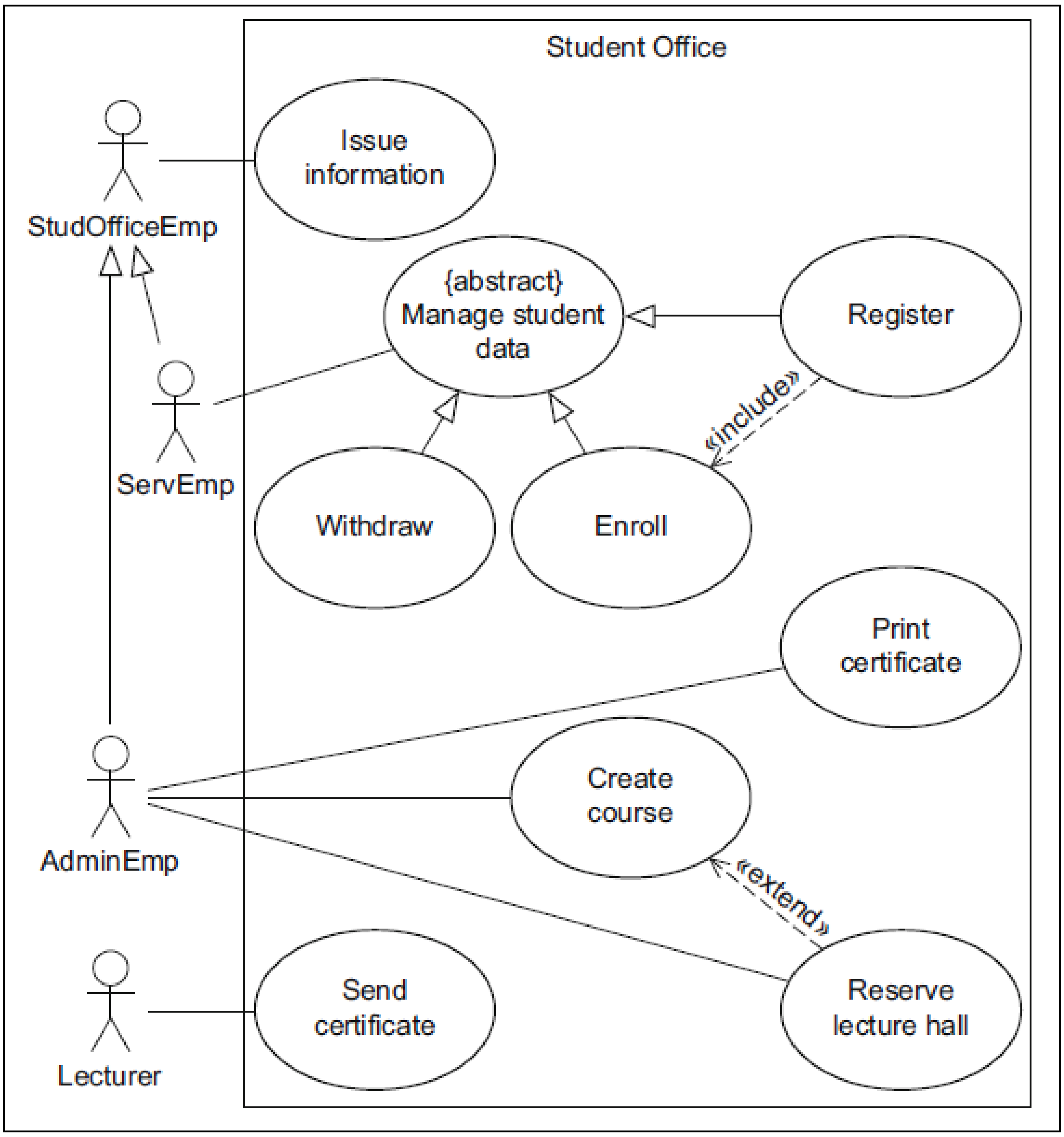
- There is a generalization relationship between Manage student data an abstract use case and Register, Enroll, and Withdraw use cases.
- There is a relationship between Reserve lecture hall and Create course use case. This relationship is that of an extension, where Reserve lecture hall extends the use case Create course.
- Register and Enroll use cases have dependency relationship, which is expressed as include relationship, where Enroll use case is included with Register use case.

Final list of Use Cases and their relationships are illustrated below:



Association between Actors and their Use Cases

We have four actors and nine use cases, where actors directly or indirectly interact with use cases as depicted below:


Student Office
StudOfficeEmp
ServEmp
AdminEmp
Lecturer
Issue information
{abstract} Manage student data
Register
«include»
Withdraw
Enroll
Print certificate
Create course
«extend»
Send certificate
Reserve lecture hall