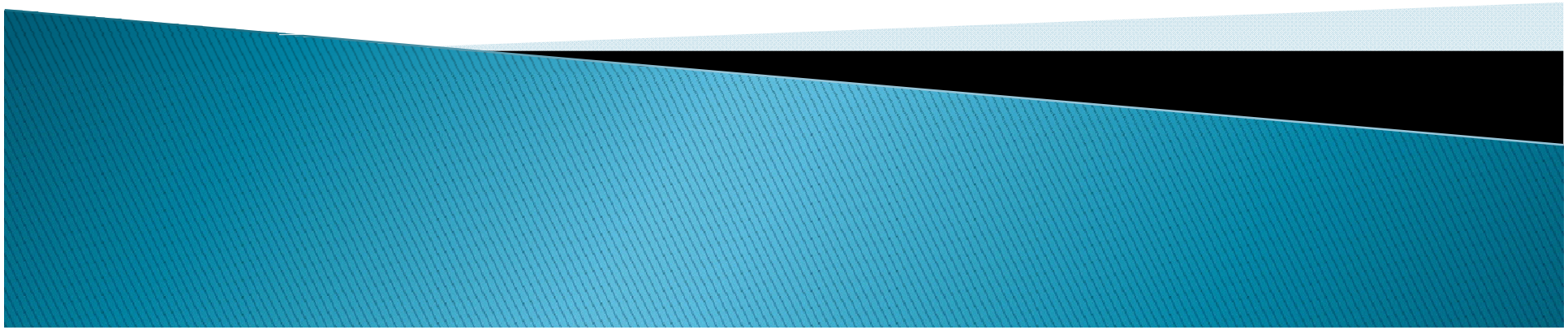
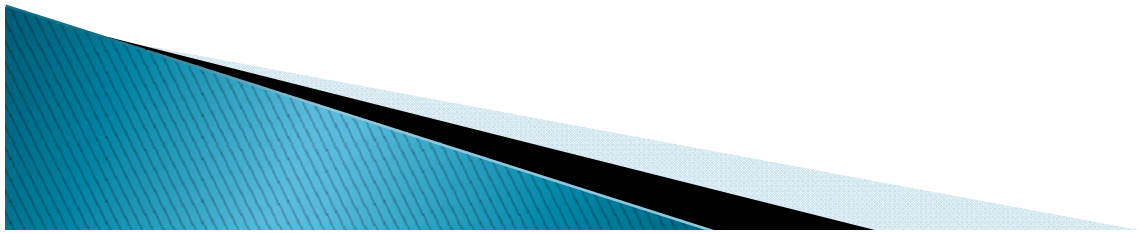
Software engineering

2010- Schedule design



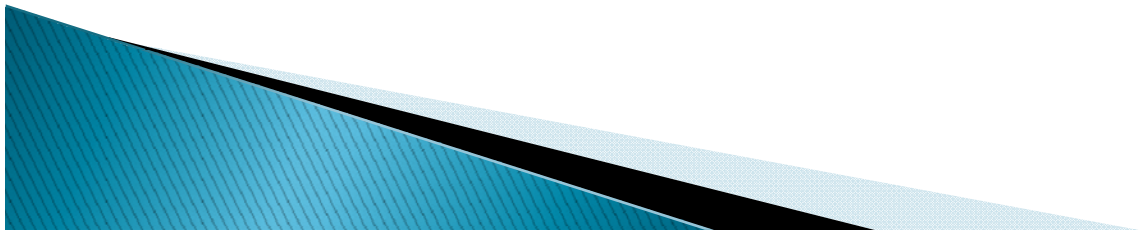
Summary

- Requirements
- Use case identification
- Use case construction
- Class construction
- Class Diagram



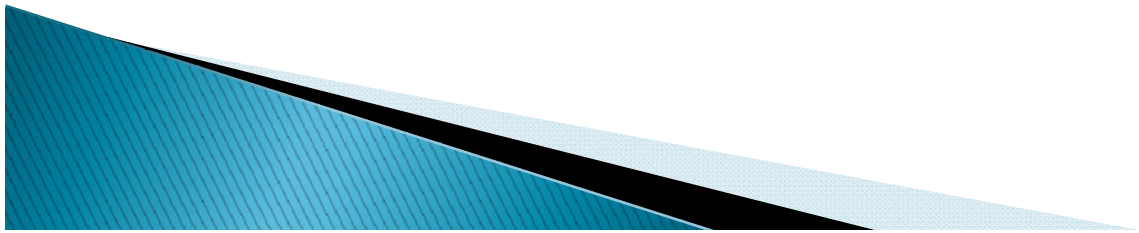
Requirements for schedule

- Consultation
  - Students
  - Professors
- Informations
  - Courses hours
  - Subject of course
  - Time of course
- Security for user
  - Id
  - Pass



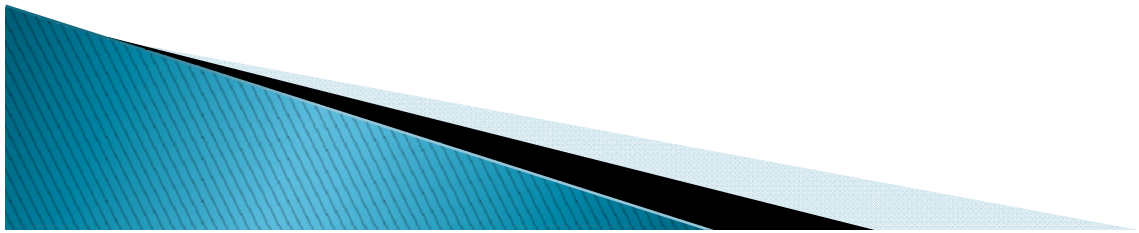
Use case identification

- Actors
  - Students
  - Professors
- System
  - Data base of university

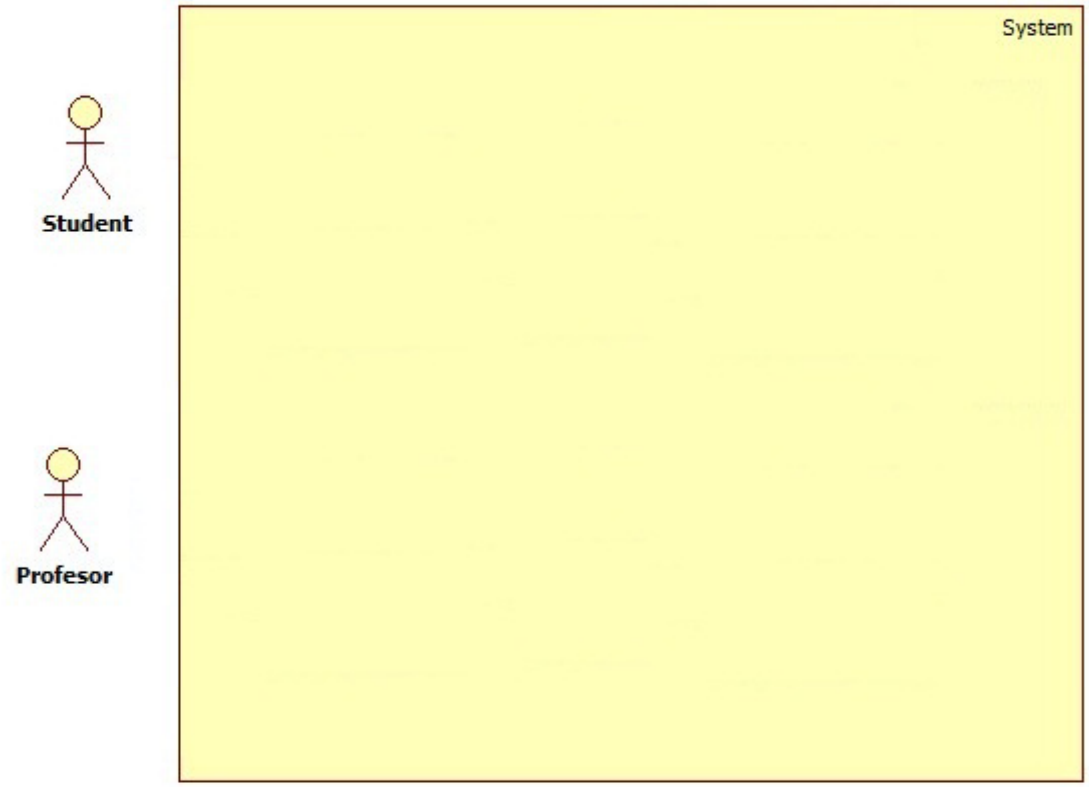


Use case identification

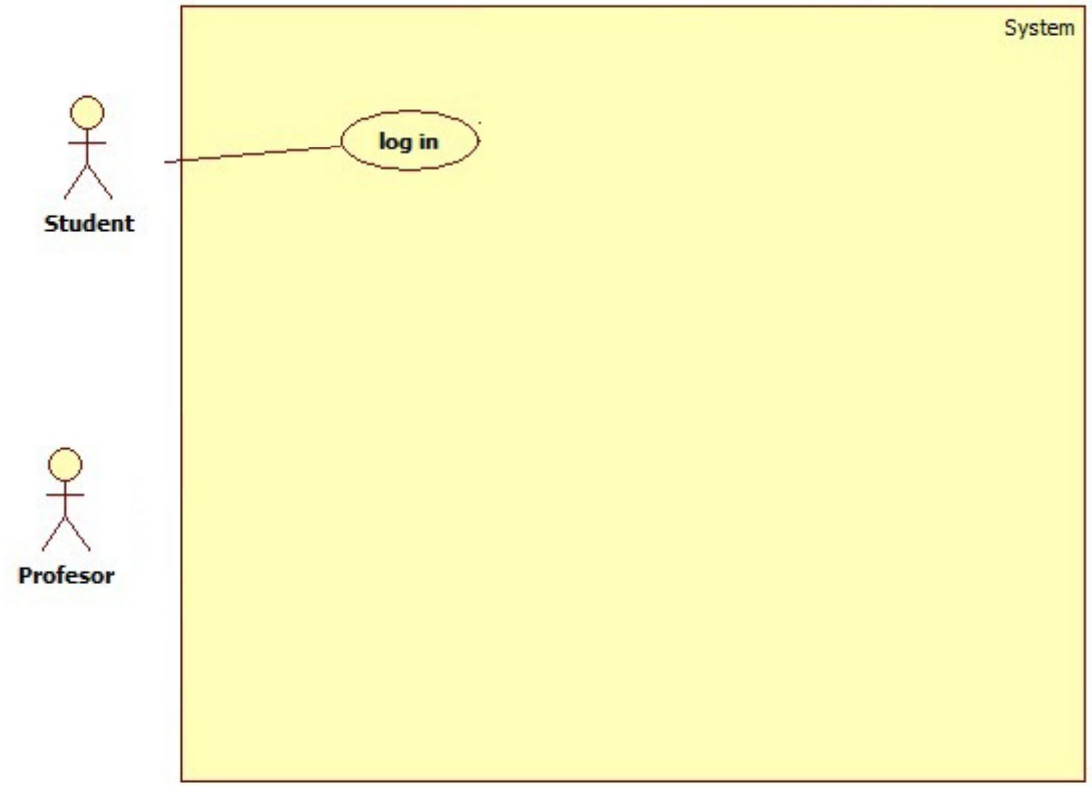
- Action
  - Each one
    - Log in/Create an account
    - Register/delete a class
    - Check class
  - Ony professor
    - Create/delete class
    - Check students



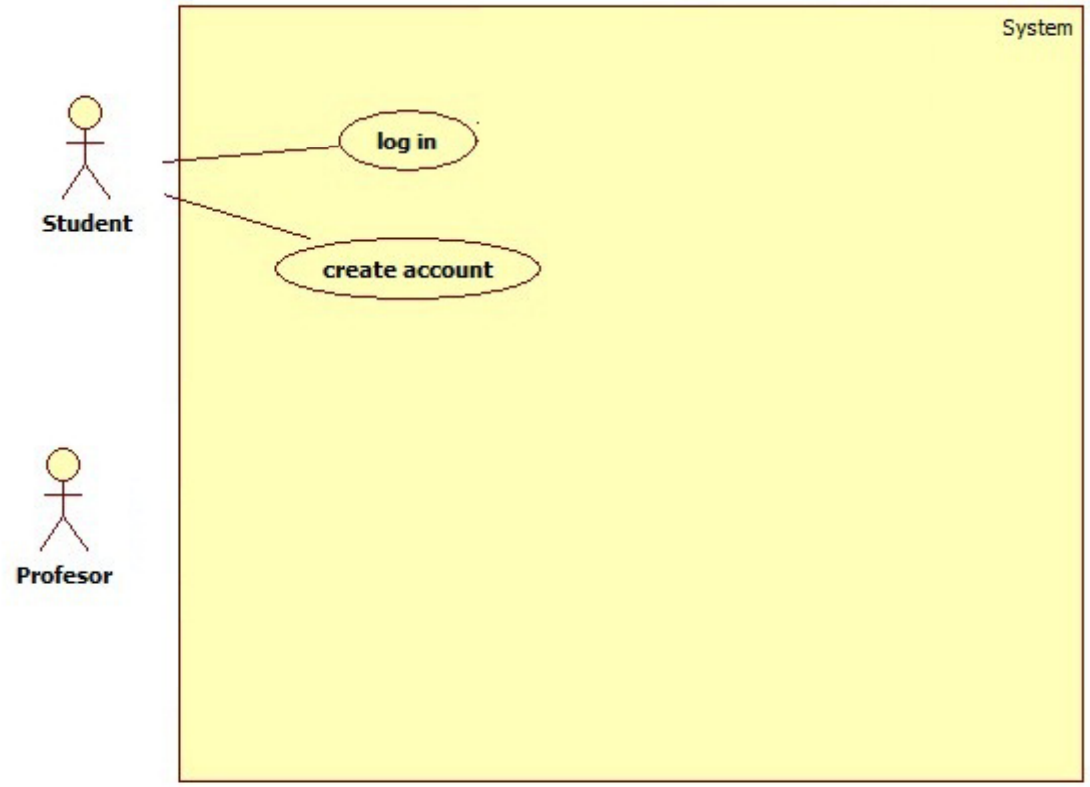
Use case construction



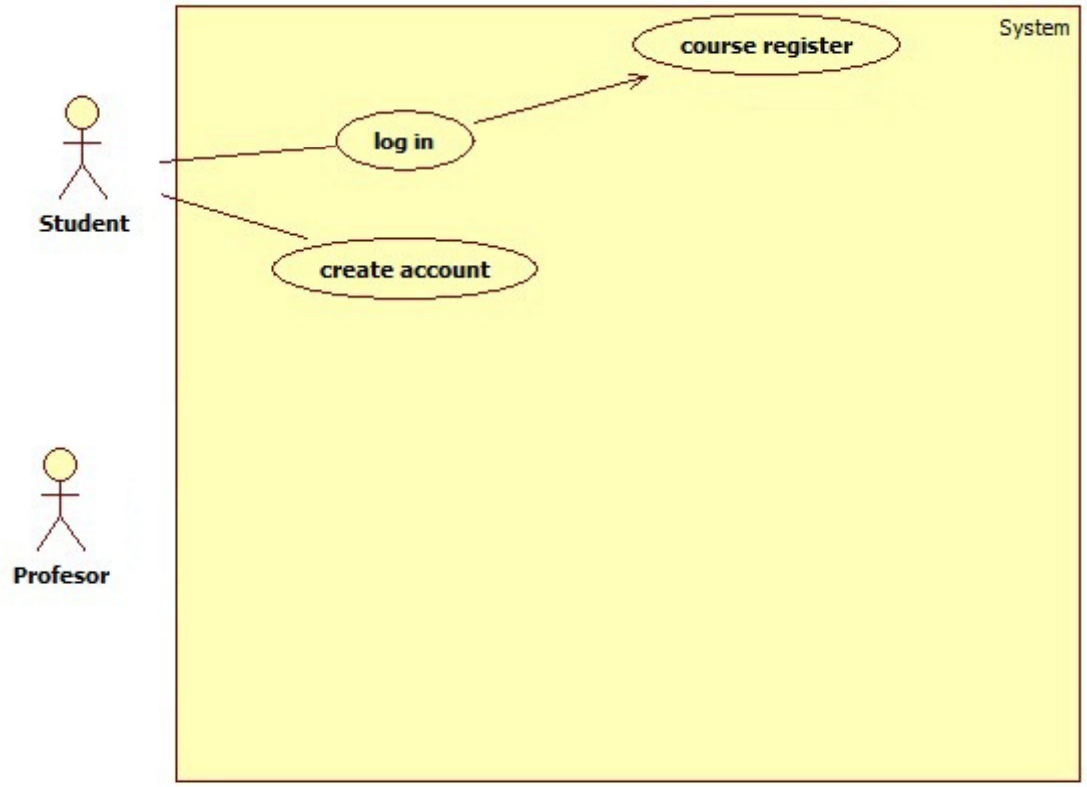
Use case construction



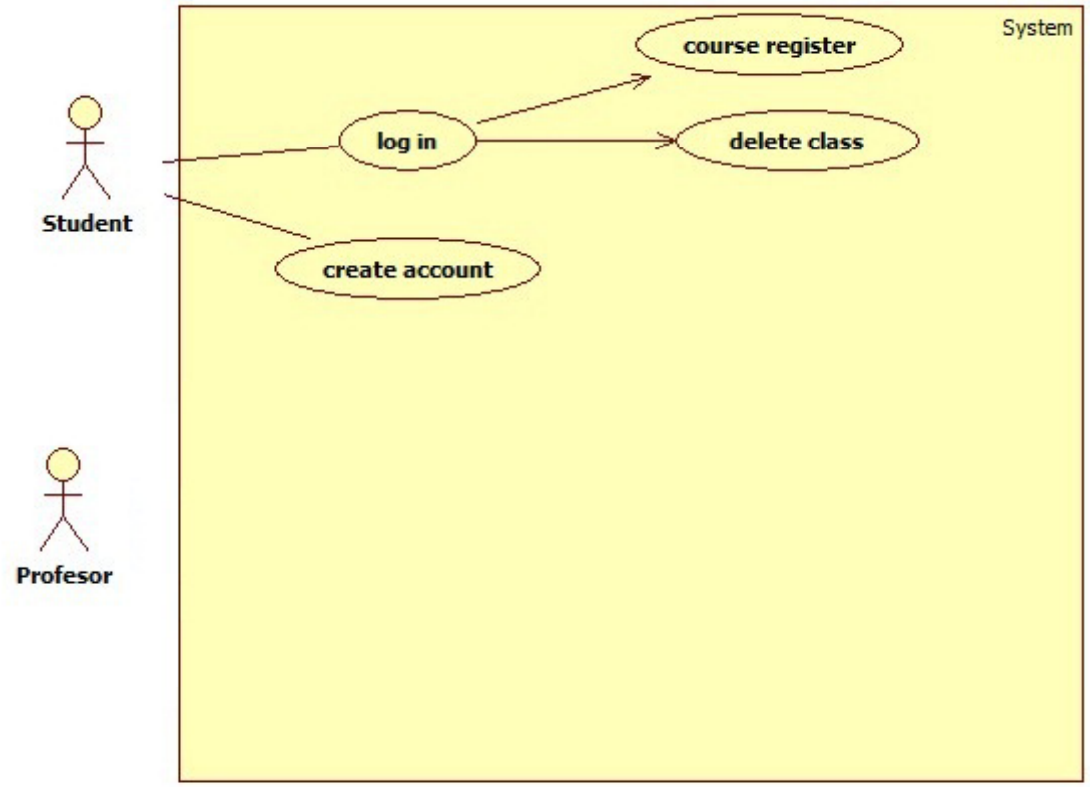
Use case construction



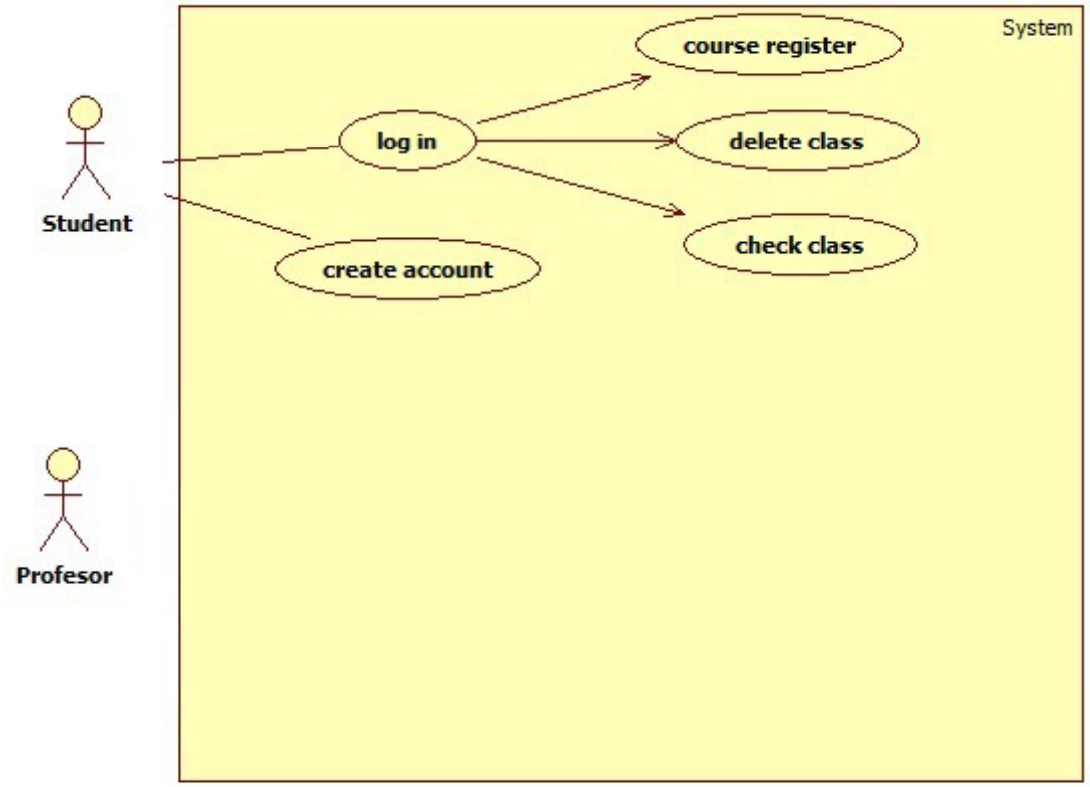
Use case construction



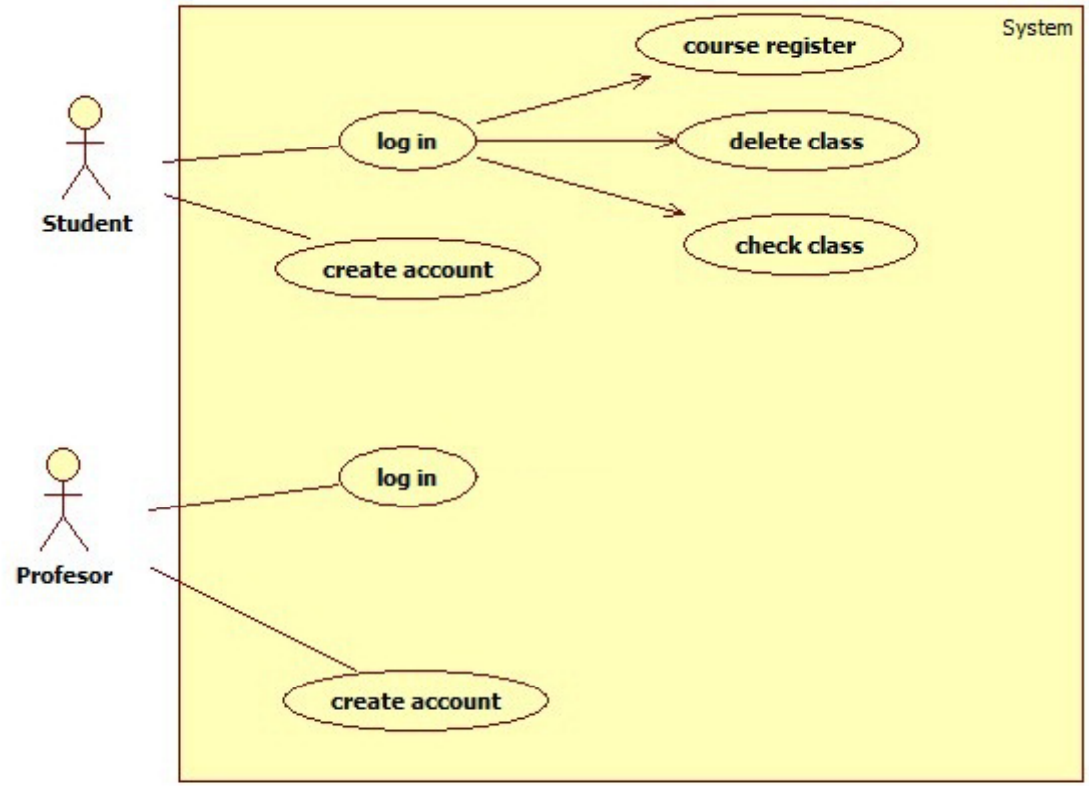
Use case construction



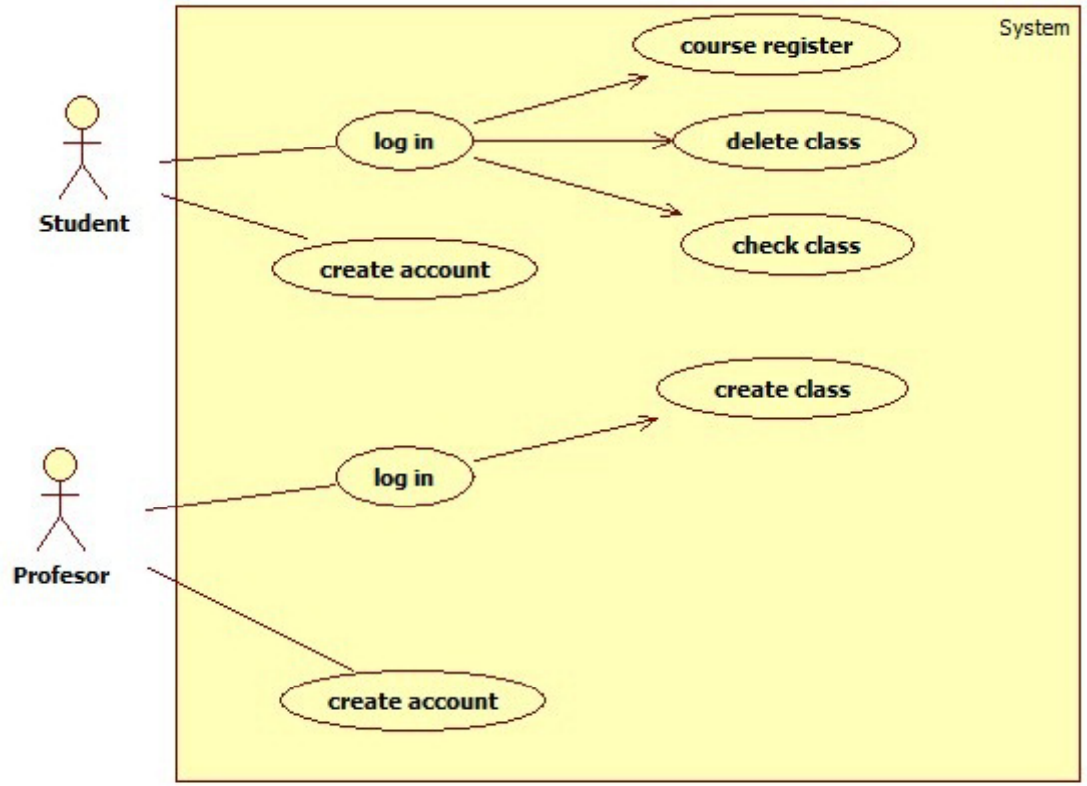
Use case construction



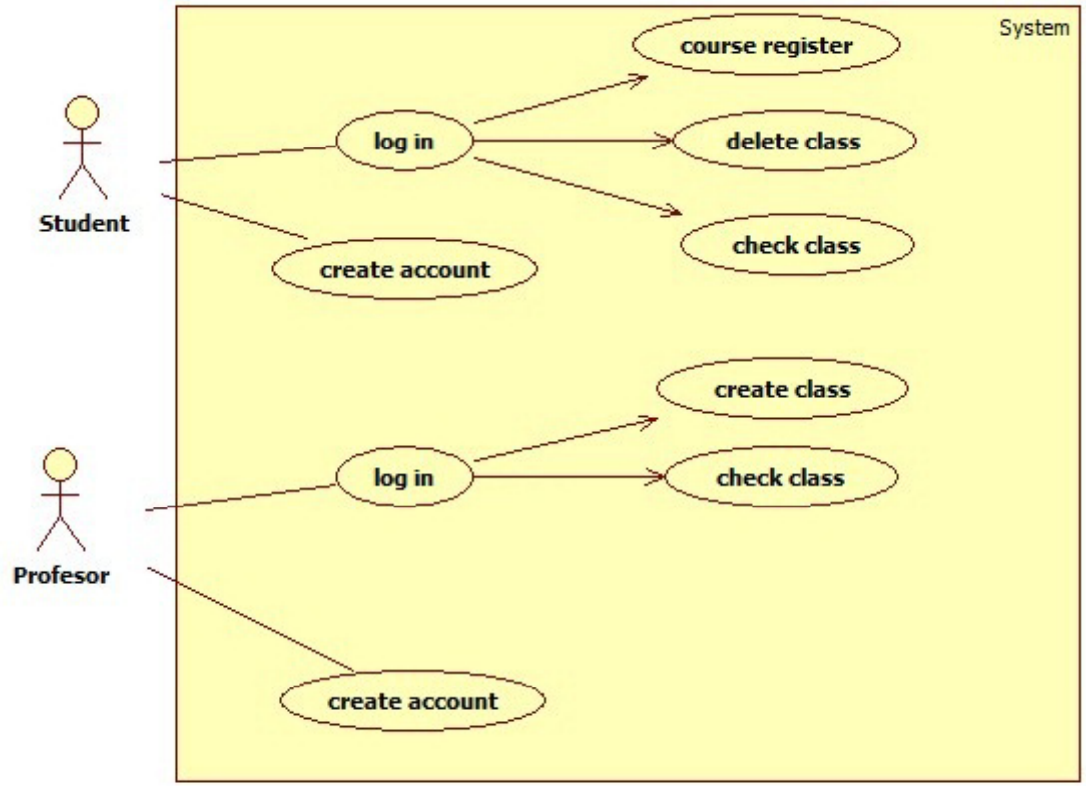
Use case construction



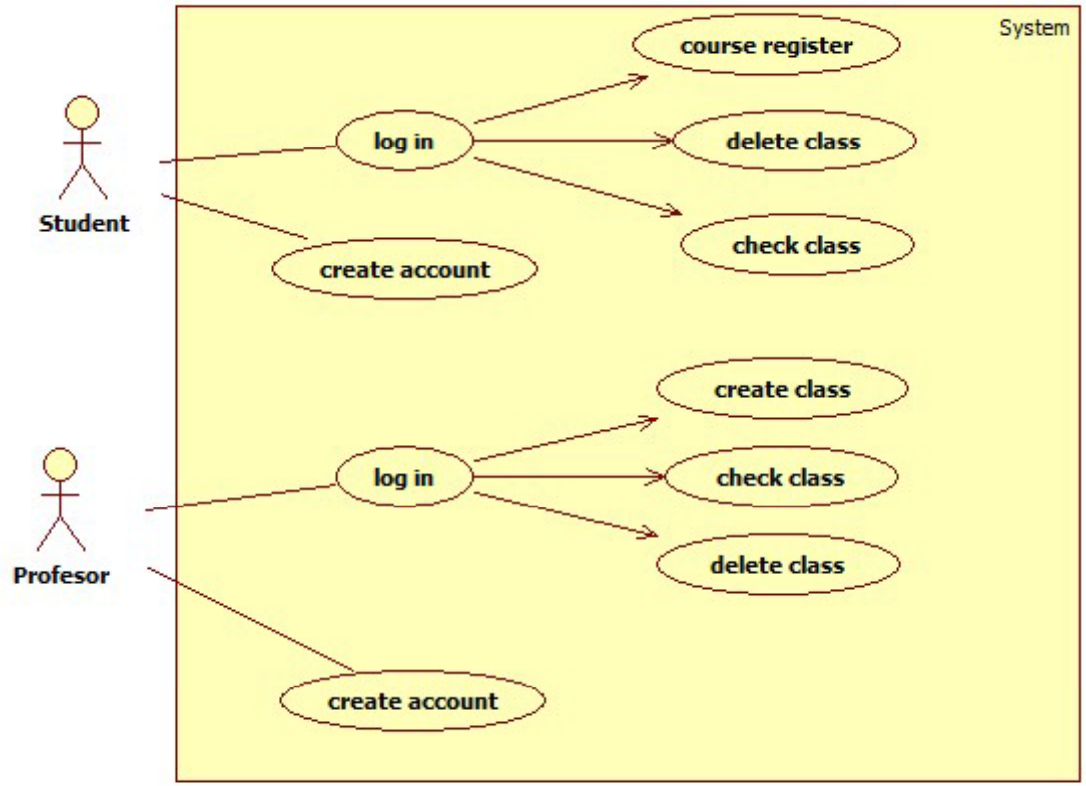
Use case construction



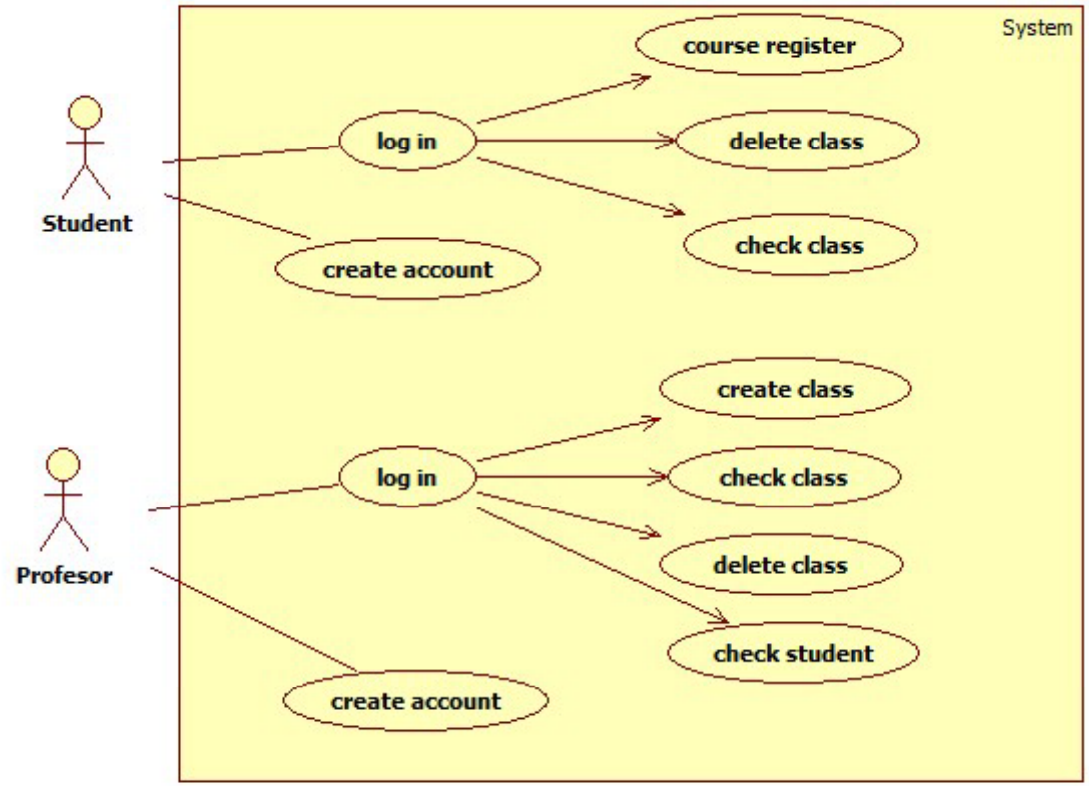
Use case construction



Use case construction

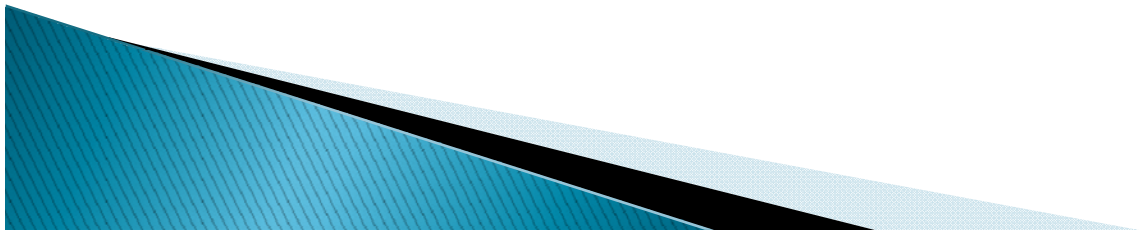


Use case construction

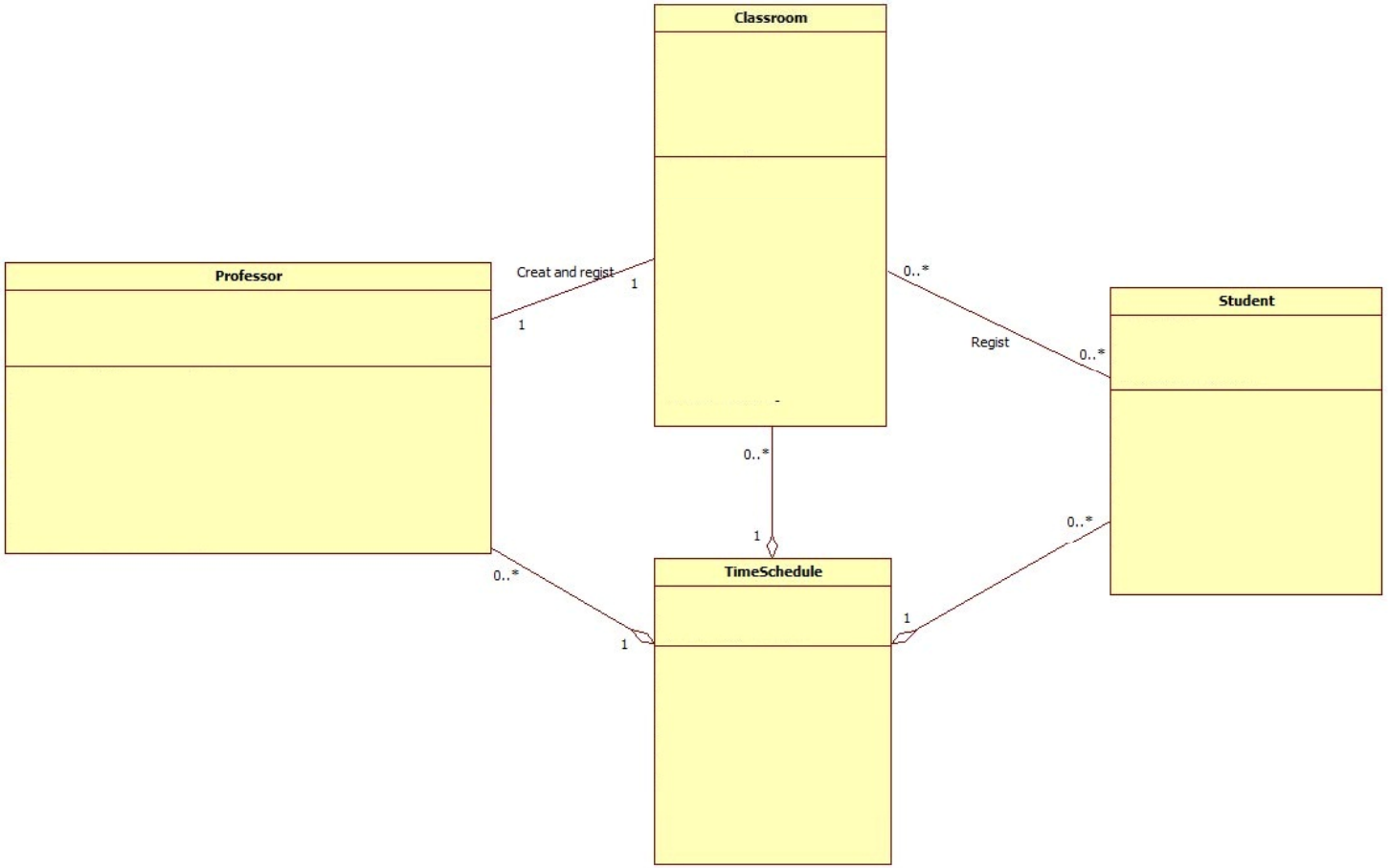


Class diagramme

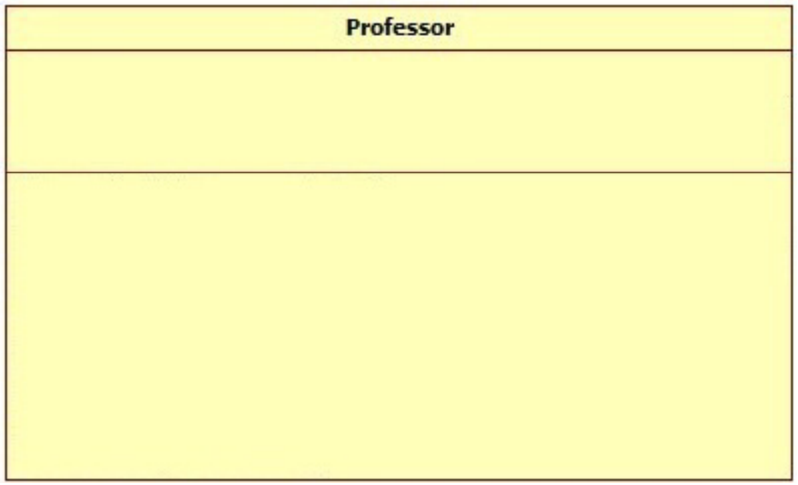
- Class name
- Attributes
- Functions



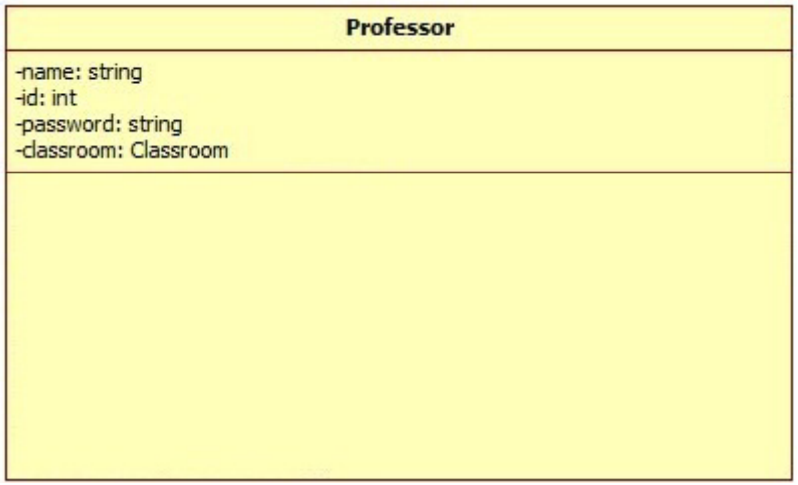
Class name


Classroom
Professor
Creat and regist
1
1
0..*
Regist
0..*
Student
0..*
1
TimeSchedule
0..*
1
1
0..*

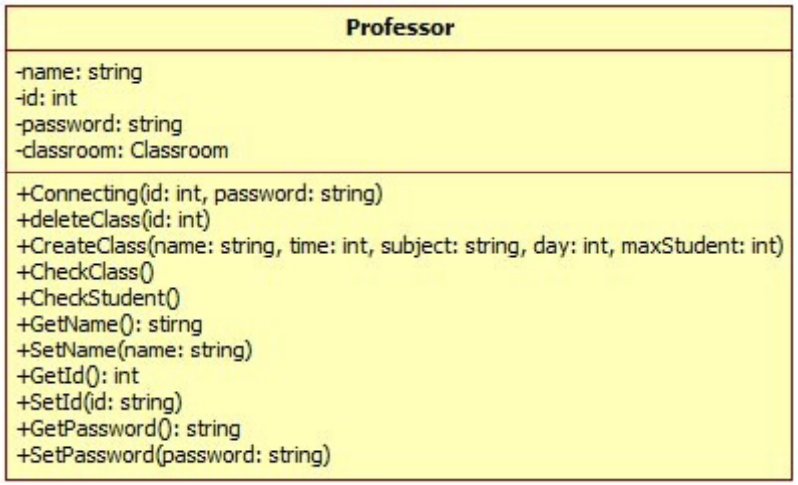
Class professor



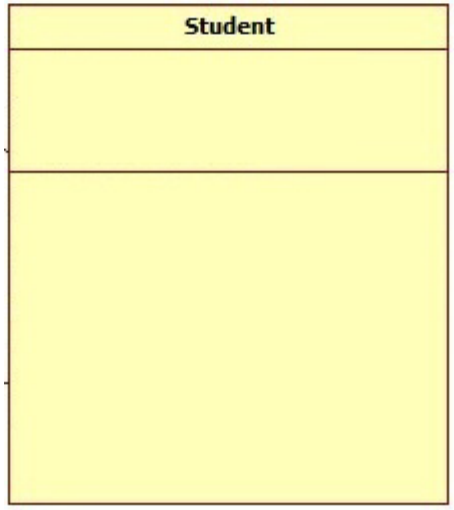
Attributes


Professor
-name: string
-id: int
-password: string
-classroom: Classroom

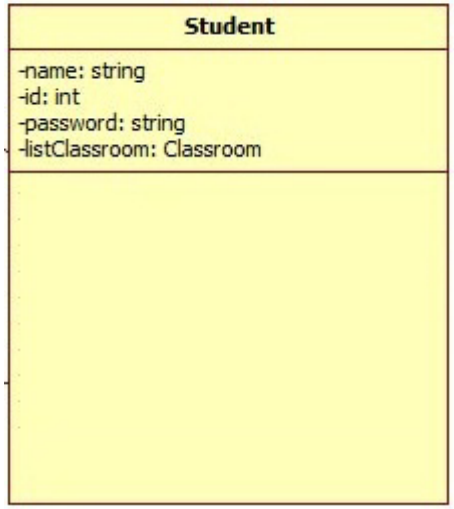
Functions


Professor
-name: string
-id: int
-password: string
-classroom: Classroom
+Connecting(id: int, password: string)
+deleteClass(id: int)
+CreateClass(name: string, time: int, subject: string, day: int, maxStudent: int)
+CheckClass()
+CheckStudent()
+GetName(): stirng
+SetName(name: string)
+GetId(): int
+SetId(id: string)
+GetPassword(): string
+SetPassword(password: string)

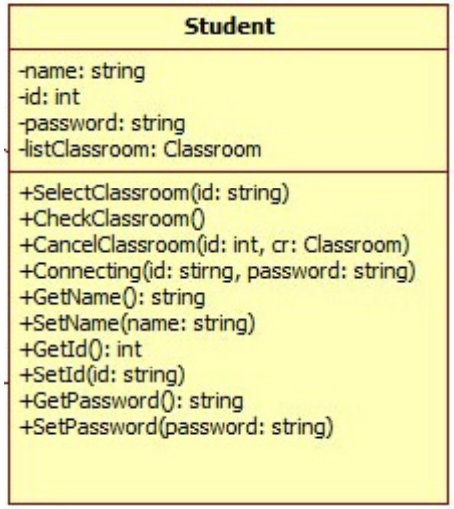
Class student



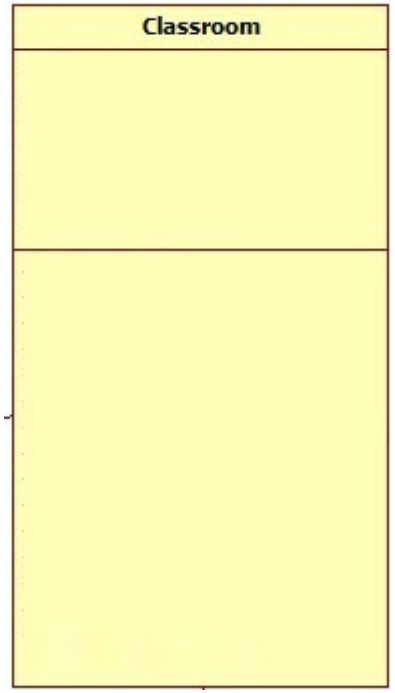
Attributes


Student
-name: string
-id: int
-password: string
-listClassroom: Classroom

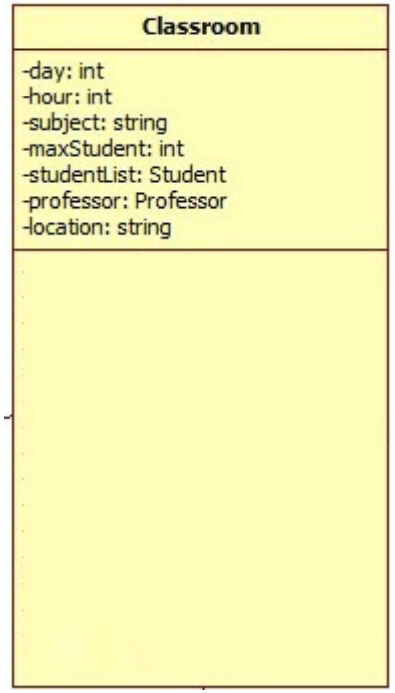
Functions


Student
-name: string
-id: int
-password: string
-listClassroom: Classroom
+SelectClassroom(id: string)
+CheckClassroom()
+CancelClassroom(id: int, cr: Classroom)
+Connecting(id: stirng, password: string)
+GetName(): string
+SetName(name: string)
+GetId(): int
+SetId(id: string)
+GetPassword(): string
+SetPassword(password: string)

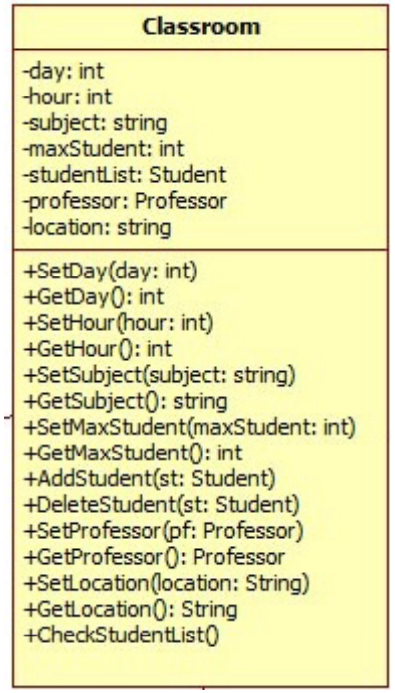
Class classroom



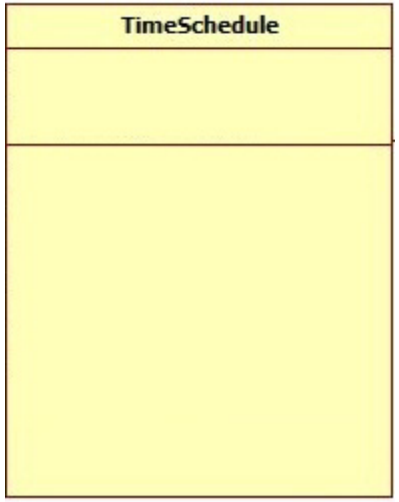
Attributes


Classroom
-day: int
-hour: int
-subject: string
-maxStudent: int
-studentList: Student
-professor: Professor
-location: string

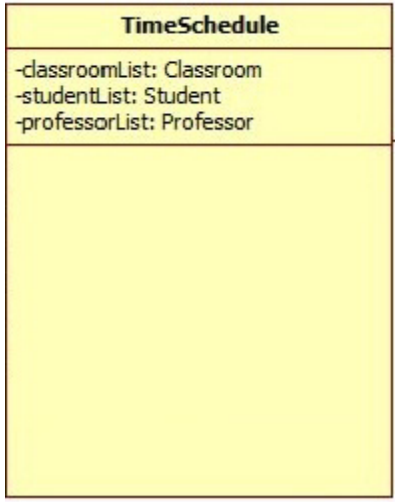
Functions


Classroom
-day: int
-hour: int
-subject: string
-maxStudent: int
-studentList: Student
-professor: Professor
-location: string
+SetDay(day: int)
+GetDay(): int
+SetHour(hour: int)
+GetHour(): int
+SetSubject(subject: string)
+GetSubject(): string
+SetMaxStudent(maxStudent: int)
+GetMaxStudent(): int
+AddStudent(st: Student)
+DeleteStudent(st: Student)
+SetProfessor(pf: Professor)
+GetProfessor(): Professor
+SetLocation(location: String)
+GetLocation(): String
+CheckStudentList()

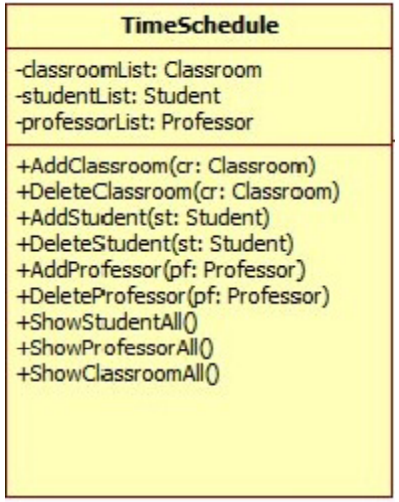
Class timeSchedule



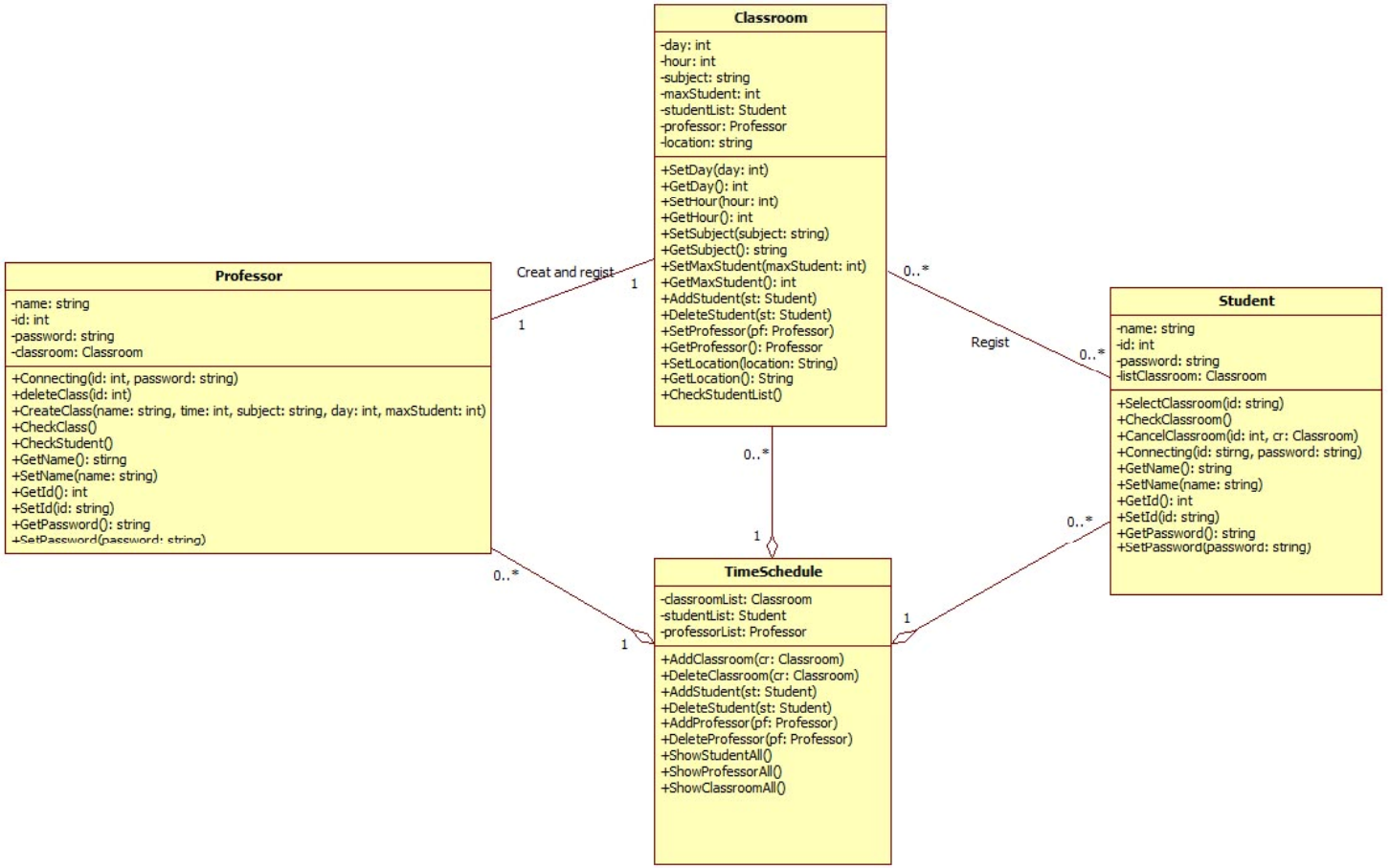
Attributes


TimeSchedule
-classroomList: Classroom
-studentList: Student
-professorList: Professor

Functions


TimeSchedule
-classroomList: Classroom
-studentList: Student
-professorList: Professor
+AddClassroom(cr: Classroom)
+DeleteClassroom(cr: Classroom)
+AddStudent(st: Student)
+DeleteStudent(st: Student)
+AddProfessor(pf: Professor)
+DeleteProfessor(pf: Professor)
+ShowStudentAll()
+ShowProfessorAll()
+ShowClassroomAll()

Class diagram



Thanks you for listening