

# DIAGRAMMA UML degli use-case

## use-case

Lo insieme omogeneo di funzionalità che un determinato gruppo può fare attore

↓ Lo e' una categoria / ruolo che un utente umano può assumere nel sistema

es. un utente può assumere più ruoli

quindi può essere rappresentato da più attori → può sia far parte

degli Studenti sia dei Docenti

c'è un caso specifico

più utenti possono essere dello stesso gruppo → Alice e Giovanni sono entrambi studenti

Un diagramma UML degli use-case è un GRAFO in cui:

- nodi sono attori e use-case
- archi che possono essere:
  1. associazione attore - use-case → a quali funzionalità può accedere un attore
  2. use case che invoca un altro use case
  3. generalizzazione tra più attori  
generalizzazione tra più usecase

### 1. attore → use-case:

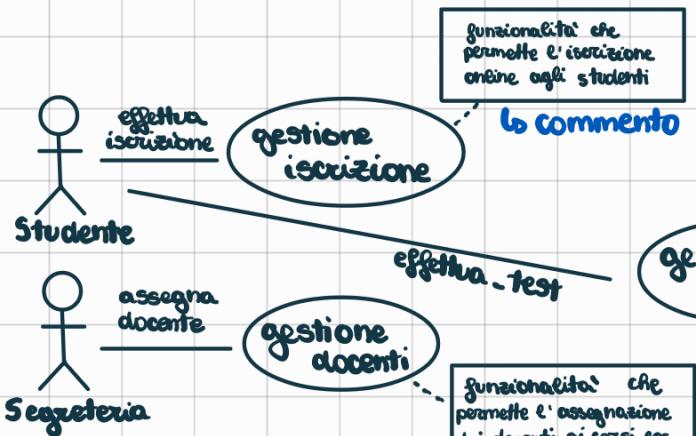
questo attore può accedere a determinate funzioni delle use-case



associazione



use-case



ogni utente che si interfaccia / accede a questa applicazione assume uno di questi tre ruoli

↓  
ruoli che possono assumere le persone

Studente → hanno il potere di iscriversi

attori → Docente → possono creare i test assegnare le soluzioni

↑ Segreteria → hanno il potere di organizzare gli orari e le lezioni dei docenti

non è detto che se ho un attore nel diagramma degli use-case allora esiste una classe di quel gruppo

nel diagramma delle classi potrebbe non esserci la classe Segreteria se non devo saperne nessuna informazione

→ lo la creiamo se dobbiamo salvare dati riguardanti Segreteria

orario apertura  
orario chiusura  
e altri

biglietti della metro → il cliente verrà rappresentato nel diagramma degli use-case ma non nel UML → quando compro un biglietto non so le informazioni del cliente

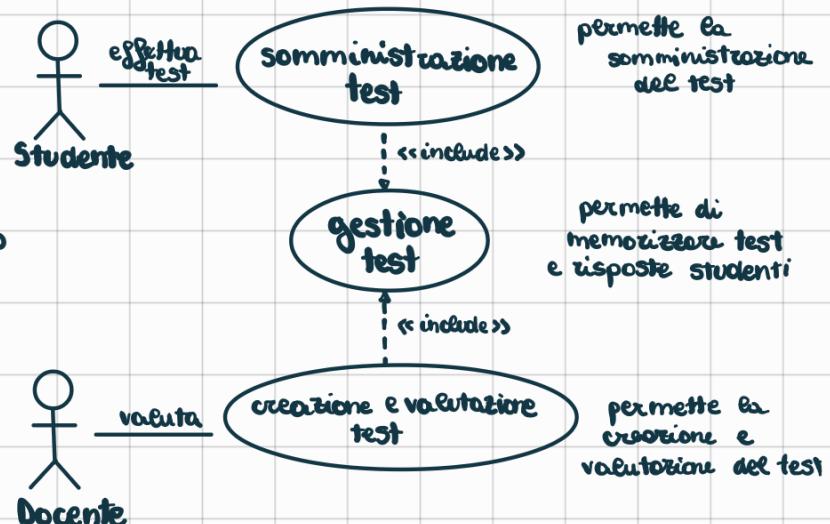
nel diagramma creato gli studenti hanno **TUTTI** i poteri di gestione test, anche di darsi un voto

## 2. use case - use case :

**INCLUSIONE** = e' una dipendenza

↳ c'e' qualche funzione nell' usecase 'Somministrazione test' che ha bisogno **sempre** di qualche funzione dello usecase 'Gestione test'

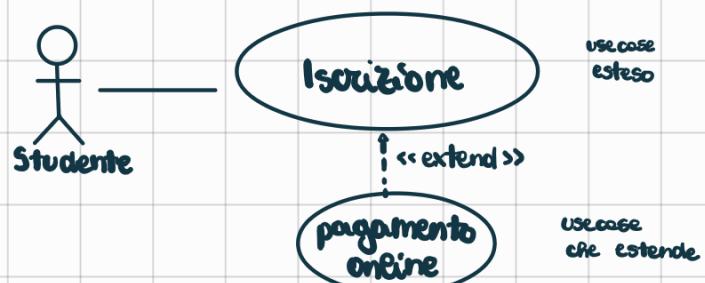
Ha **sempre bisogno** di un'altro usecase



Docente crea un test e grazie a gestione test lo salva, Studente svolge il test e le sue risposte vengono salvate grazie a gestione test → qui **serve** questa classe

## ESTENSIONE

↳ alcune funzioni dell'use case "Iscrizione", **solo in alcuni casi**, vengono estese dall'usecase "Pagamento online"



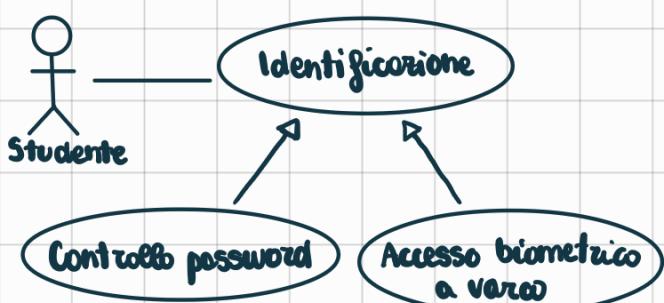
normalmente uno studente puo' iscriversi senza nessun problema e SE VUOLE (non sempre) c'e' l'opzione per pagare online  
↳ se qualcosa di particolare accade (sceglie di pagare online) uso le funzionalita' di un'altro usecase **estendendo** quello principale

## 3. GENERALIZZAZIONE

### • tra usecase

↳ alcune funzioni di "Identificazione" vengono **rimpiazzate** dagli use case sottostanti:

posso identificarmi in uno dei due modi



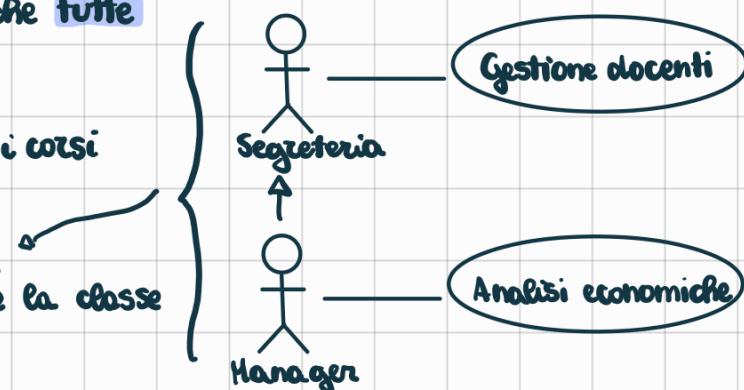
non usiamo {disjoint} o {complete}  
ne ↑ ↳ non nei diagrammi use-case

### • tra attori

l'attore 'Manager' puo' fare le veci dell'attore 'Segreteria' ottenendo anche **tutte le sue funzionalita'**

↳ le eredita e puo' anch'esso gestire i corsi dei docenti, come la segreteria

**NON IMPLICA** che questi attori esistano nel diagramma delle classi, ne' tantomeno che la classe Manager sia una sottoclasse di Segreteria



Il diagramma degli use-case mostra:

- quali attori possono usare il sistema
- quali macro-funzionalità sono accessibili ai diversi attori

Ogni use-case del diagramma è affiancato da un documento di specifica

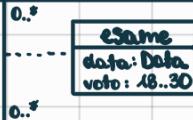


TIPICAMENTI cambiano  
gli oggetti estensionalmente

non c'è un  
oggetto che  
lo invoca  
'this'

Studente
matricola: Intero > 0 {id}
nome: Stringa

media\_fino\_a(d: Data): reale in 18..30



Corso
nome: Stringa
voto_medio(): Reale in 18..30



operation che puo' fare

Specifiche dello use-case Verbalizzazione

verbalizza (s: Studente, c: Corso, d: Data, v: 18..30)

precondizioni:

- (s, c) non è già un link "esame"

postcondizioni:

- viene creato il link (s, c) di tipo "esame", con valori "d" e "v" per gli attributi "data" e "voto"

Se ci sono altre operazioni dello use-case le metto

infatti nel diagramma delle classi non esiste la classe Docente perché non vogliamo in questo caso salvare le sue informazioni