



UNIVERSITÀ  
DEGLI STUDI  
DI BERGAMO

Dipartimento  
di Ingegneria Gestionale,  
dell'Informazione e della Produzione

# Tutorato UML 1

Ingegneria del Software

*Nico Pellegrinelli*

[nico.pellegrinelli@unibg.it](mailto:nico.pellegrinelli@unibg.it)

# **Esercizio completo**

# **SUBMISSION SYSTEM**



**UNIVERSITÀ  
DEGLI STUDI  
DI BERGAMO**

Dipartimento  
di Ingegneria Gestionale,  
dell'Informazione e della Produzione

# Testo dell'esercizio

Ogni corso inserito nel sistema ha dei professori associati. L'inserimento viene svolto dall'amministratore di un corso, che è anch'esso un professore. Come parte di un corso, i professori possono creare tasks e valutare i lavori sottomessi dagli studenti. Quindi, il professore può assegnare dei punti ad ogni lavoro e fornire un feedback.

L'amministratore del corso definisce quali professori possono valutare quali specifici lavori. Alla fine del corso, l'amministratore del corso provvede al rilascio dei certificati. Il voto di uno studente viene calcolato in base al numero totale di punti per i lavori sottomessi.

Gli studenti possono seguire corsi e caricare lavori.

Tutti gli utenti – studenti e professori – possono gestire i loro dati, vedere i corsi e l'insieme di task per ogni corso (ammesso che l'utente sia coinvolto nel corso), e vedere i lavori sottomessi e i punti di valutazione. Ogni studente può vedere solamente i suoi lavori ed i suoi voti. Ogni professore può vedere i lavori assegnati a lui e i voti di questi lavori. L'amministratore del corso, invece, ha diritto di accesso a tutti i dati.

Un corso viene creato o eliminato da un impiegato amministratore. Quando il corso viene creato, almeno un amministratore deve essere inserito. Ulteriori amministratori possono essere aggiunti o eliminati in un secondo momento. L'amministratore può anche cancellare interi corsi.

Le informazioni riguardo gli utenti vengono trasferite automaticamente da un altro sistema, quindi le funzioni che permettono la creazione dei dati degli utenti non sono necessarie.

Tutte le funzioni del sistema possono essere usate solamente dalle persone che sono loggiate.



# Use Case Diagram

Ogni corso inserito nel sistema ha dei professori associati. L'inserimento viene svolto dall'amministratore di un corso, che è anch'esso un professore. Come parte di un corso, i professori possono creare tasks e valutare i lavori sottomessi dagli studenti. Quindi, il professore può assegnare dei punti ad ogni lavoro e fornire un feedback.

L'amministratore del corso definisce quali professori possono valutare quali specifici lavori. Alla fine del corso, l'amministratore del corso provvede al rilascio dei certificati. Il voto di uno studente viene calcolato in base al numero totale di punti per i lavori sottomessi.

Gli studenti possono seguire corsi e caricare lavori.

Tutti gli utenti – studenti e professori – possono gestire i loro dati, vedere i corsi e l'insieme di task per ogni corso (ammesso che l'utente sia coinvolto nel corso), e vedere i lavori sottomessi e i punti di valutazione. Ogni studente può vedere solamente i suoi lavori ed i suoi voti. Ogni professore può vedere i lavori assegnati a lui e i voti di questi lavori. L'amministratore del corso, invece, ha diritto di accesso a tutti i dati.

Un corso viene creato o eliminato da un impiegato amministratore. Quando il corso viene creato, almeno un amministratore deve essere inserito. Ulteriori amministratori possono essere aggiunti o eliminati in un secondo momento. L'amministratore può anche cancellare interi corsi.

Le informazioni riguardo gli utenti vengono trasferite automaticamente da un altro sistema, quindi le funzioni che permettono la creazione dei dati degli utenti non sono necessarie.

Tutte le funzioni del sistema possono essere usate solamente dalle persone che sono loggate.



# Use Case Diagram - Attori

Ogni corso inserito nel sistema ha dei **professori** associati. L'inserimento viene svolto dall'**amministratore** di un corso, che è anch'esso un professore. Come parte di un corso, i professori possono creare tasks e valutare i lavori sottomessi dagli **studenti**. Quindi, il professore può assegnare dei punti ad ogni lavoro e fornire un feedback.

L'amministratore del corso definisce quali professori possono valutare quali specifici lavori. Alla fine del corso, l'amministratore del corso provvede al rilascio dei certificati. Il voto di uno studente viene calcolato in base al numero totale di punti per i lavori sottomessi.

Gli studenti possono seguire corsi e caricare lavori.

Tutti gli **utenti** – studenti e professori – possono gestire i loro dati, vedere i corsi e l'insieme di task per ogni corso (ammesso che l'utente sia coinvolto nel corso), e vedere i lavori sottomessi e i punti di valutazione. Ogni studente può vedere solamente i suoi lavori ed i suoi voti. Ogni professore può vedere i lavori assegnati a lui e i voti di questi lavori. L'amministratore del corso, invece, ha diritto di accesso a tutti i dati.

Un corso viene creato o eliminato da un **impiegato amministratore**. Quando il corso viene creato, almeno un amministratore deve essere inserito. Ulteriori amministratori possono essere aggiunti o eliminati in un secondo momento. L'amministratore può anche cancellare interi corsi.

Le informazioni riguardo gli utenti vengono trasferite automaticamente da un altro sistema, quindi le funzioni che permettono la creazione dei dati degli utenti non sono necessarie.

Tutte le funzioni del sistema possono essere usate solamente dalle persone che sono loggate.



# Use Case Diagram – Relazioni tra Attori

Ogni corso inserito nel sistema ha dei **professori** associati. L'inserimento viene svolto dall'**amministratore** di un corso, che è anch'esso un professore. Come parte di un corso, i professori possono creare tasks e valutare i lavori sottomessi dagli **studenti**. Quindi, il professore può assegnare dei punti ad ogni lavoro e fornire un feedback.

L'amministratore del corso definisce quali professori possono valutare quali specifici lavori. Alla fine del corso, l'amministratore del corso provvede al rilascio dei certificati. Il voto di uno studente viene calcolato in base al numero totale di punti per i lavori sottomessi.

Gli studenti possono seguire corsi e caricare lavori.

Tutti gli **utenti** – **studenti e professori** – possono gestire i loro dati, vedere i corsi e l'insieme di task per ogni corso (ammesso che l'utente sia coinvolto nel corso), e vedere i lavori sottomessi e i punti di valutazione. Ogni studente può vedere solamente i suoi lavori ed i suoi voti. Ogni professore può vedere i lavori assegnati a lui e i voti di questi lavori. L'amministratore del corso, invece, ha diritto di accesso a tutti i dati.

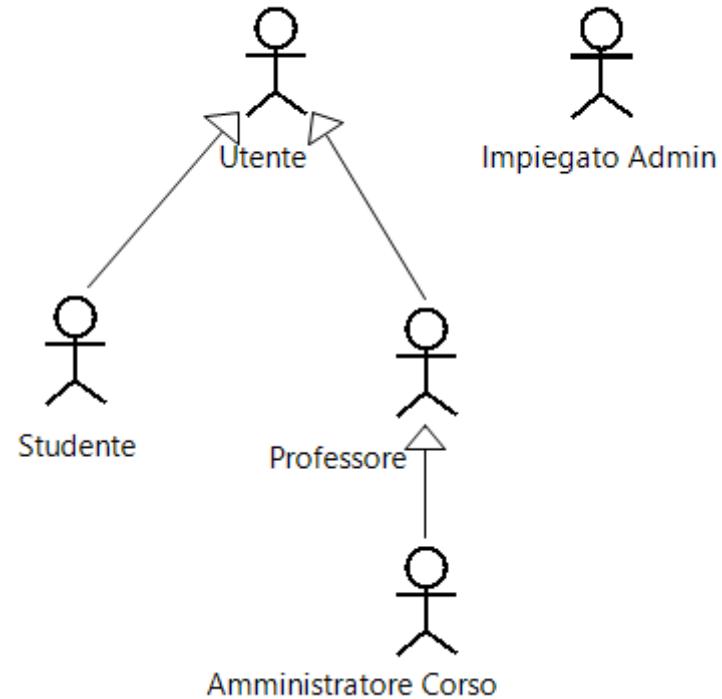
Un corso viene creato o eliminato da un **impiegato amministratore**. Quando il corso viene creato, almeno un amministratore deve essere inserito. Ulteriori amministratori possono essere aggiunti o eliminati in un secondo momento. L'amministratore può anche cancellare interi corsi.

Le informazioni riguardo gli utenti vengono trasferite automaticamente da un altro sistema, quindi le funzioni che permettono la creazione dei dati degli utenti non sono necessarie.

Tutte le funzioni del sistema possono essere usate solamente dalle persone che sono loggate.



# Use Case Diagram -Attori



# Use Case Diagram – Casi d'uso

?

Ogni corso inserito nel sistema ha dei **professori** associati. L'**inserimento** viene svolto dall'**amministratore** di un corso, che è anch'esso un professore. Come parte di un corso, i professori possono **creare tasks** e **valutare i lavori** sottomessi dagli **studenti**. Quindi, il professore può **assegnare dei punti** ad ogni lavoro e fornire un feedback.

L'amministratore del corso **definisce quali professori possono valutare quali specifici lavori**. Alla fine del corso, l'amministratore del corso provvede al **rilascio dei certificati**. Il voto di uno studente viene calcolato in base al numero totale di punti per i lavori sottomessi.

Gli studenti possono **seguire corsi** e **caricare lavori**.

Tutti gli **utenti** – studenti e professori – possono **gestire i loro dati**, **vedere i corsi** e **l'insieme di task** per ogni corso (ammesso che l'utente sia coinvolto nel corso), e **vedere i lavori sottomessi e i punti di valutazione**. Ogni studente può vedere solamente i suoi lavori ed i suoi voti. Ogni professore può vedere i lavori assegnati a lui e i voti di questi lavori. L'amministratore del corso, invece, ha diritto di accesso a tutti i dati.

**Un corso viene creato** o **eliminato** da un **impiegato amministratore**. Quando il corso viene creato, almeno **un amministratore deve essere inserito**. Ulteriori amministratori possono essere **aggiunti o eliminati** in un secondo momento. L'amministratore può anche cancellare interi corsi.

Le informazioni riguardo gli utenti vengono trasferite automaticamente da un altro sistema, quindi le funzioni che permettono la creazione dei dati degli utenti non sono necessarie.

Tutte le funzioni del sistema possono essere usate solamente dalle persone che sono loggate.



# Use Case Diagram – Relazioni tra Casi d'uso

Ogni corso inserito nel sistema ha dei **professori** associati. L'**inserimento** viene svolto dall'**amministratore** di un corso, che è anch'esso un professore. Come parte di un corso, i professori possono **creare tasks** e **valutare i lavori** sottomessi dagli **studenti**. Quindi, il professore può **assegnare dei punti** ad ogni lavoro e fornire un feedback.

L'amministratore del corso **definisce quali professori possono valutare quali specifici lavori**. Alla fine del corso, l'amministratore del corso provvede al **rilascio dei certificati**. Il voto di uno studente viene calcolato in base al numero totale di punti per i lavori sottomessi.

Gli studenti possono **seguire corsi** e **caricare lavori**.

Tutti gli **utenti** – studenti e professori – possono **gestire i loro dati**, **vedere i corsi** e **l'insieme di task** per ogni corso (ammesso che l'utente sia coinvolto nel corso), e **vedere i lavori sottomessi e i punti di valutazione**. Ogni studente può vedere solamente i suoi lavori ed i suoi voti. Ogni professore può vedere i lavori assegnati a lui e i voti di questi lavori. L'amministratore del corso, invece, ha diritto di accesso a tutti i dati.

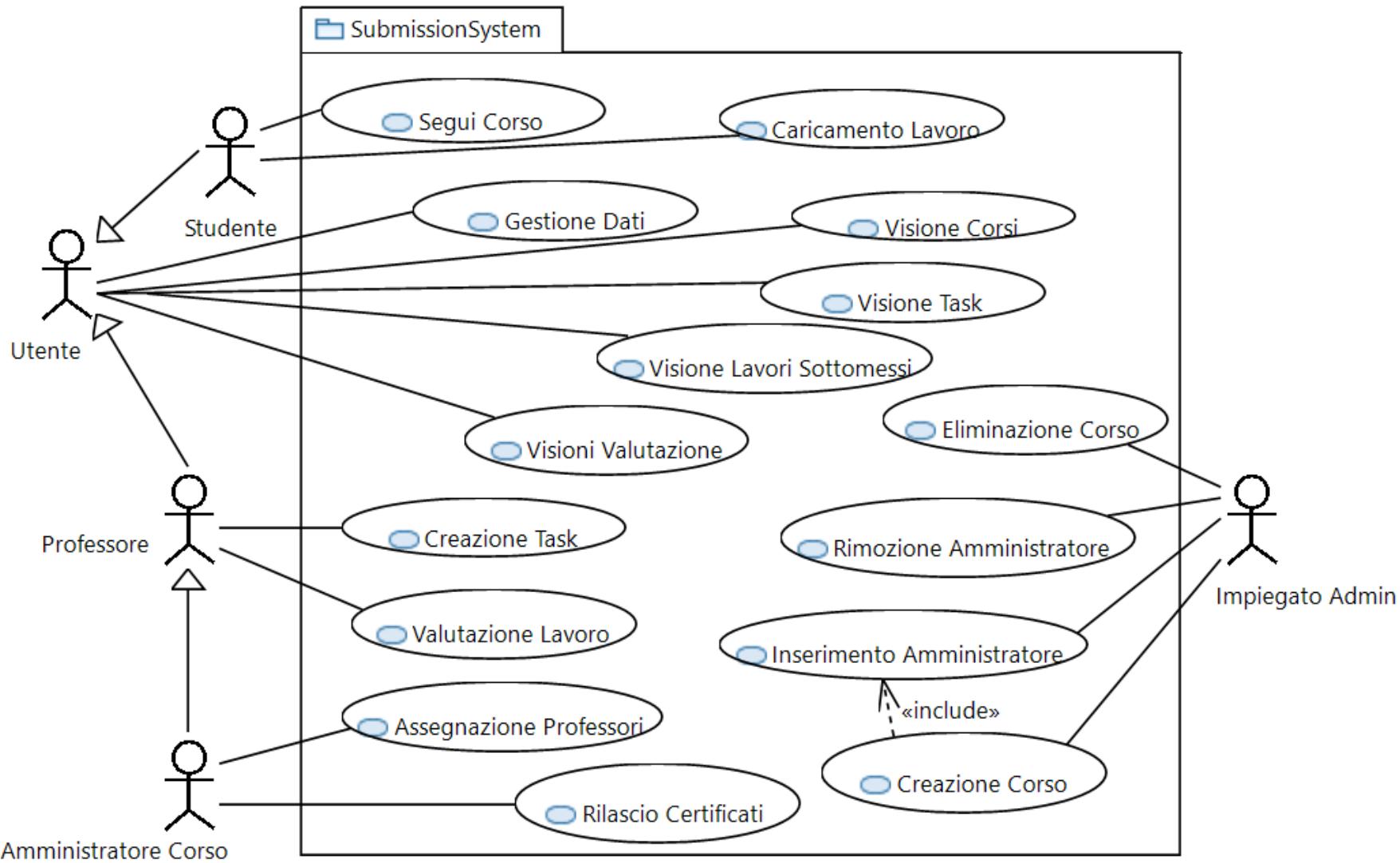
**Un corso viene creato** o **eliminato** da un **impiegato amministratore**. Quando il corso viene creato, almeno **un amministratore deve essere inserito**. Ulteriori amministratori possono essere **aggiunti o eliminati** in un secondo momento. L'amministratore può anche cancellare interi corsi.

Le informazioni riguardo gli utenti vengono trasferite automaticamente da un altro sistema, quindi le funzioni che permettono la creazione dei dati degli utenti non sono necessarie.

Tutte le funzioni del sistema possono essere usate solamente dalle persone che sono loggate.



# Use Case Diagram -Soluzione



# Use Case Diagram – Descrizione casi d'uso

Identificare per il caso d'suo VISIONE TASK:

- Nome
- Breve descrizione
- Pre-condizioni
- Post-condizioni
- Situazioni di errore
- Stato del sistema nel caso in cui ci sia un errore
- Attori
- Trigger
- Processo standard
- Processo alternativo



# Use Case Diagram – Descrizione casi d'uso

Caso d'suo VISIONE TASK:

- **Nome**

Visualizzazione di un task

- **Breve descrizione**

L'utente visualizza le informazioni riguardanti un task relativo ad un corso

- **Pre-condizioni**

L'utente è coinvolto nel corso

- **Post-condizioni**

L'utente ha visualizzato tutte le informazioni del task, in funzione del suo ruolo (studente, professore, amministratore)

- **Situazioni di errore**

L'utente seleziona un corso in cui non è coinvolto

- **Stato del sistema nel caso in cui ci sia un errore**

Il sistema ritorna alla pagina iniziale

- **Attori**

Utente



# Use Case Diagram – Descrizione casi d'uso

- **Trigger**

L'utente richiede di visualizzare le informazioni riguardanti un corso e poi quelle relative ad un task

- **Processo standard**

1. L'utente fa il login nel sistema
2. Il sistema verifica il livello di autorizzazione dell'utente (studente, professore, amministratore)
3. L'utente seleziona l'opzione "Seleziona Corso"
4. L'utente seleziona il corso da visualizzare
5. L'utente seleziona l'opzione "Visualizza Task"
6. L'utente seleziona il task da visualizzare
7. Il sistema mostra le informazioni del task in funzione del livello di autorizzazione dell'utente

- **Processo alternativo**

5. Il sistema notifica all'utente che non è coinvolto in quel corso e ritorna alla pagina iniziale



# Class Diagram

Ogni corso inserito nel sistema ha dei professori associati. L'inserimento viene svolto dall'amministratore di un corso, che è anch'esso un professore. Come parte di un corso, i professori possono creare tasks e valutare i lavori sottomessi dagli studenti. Quindi, il professore può assegnare dei punti ad ogni lavoro e fornire un feedback.

L'amministratore del corso definisce quali professori possono valutare quali specifici lavori. Alla fine del corso, l'amministratore del corso provvede al rilascio dei certificati. Il voto di uno studente viene calcolato in base al numero totale di punti per i lavori sottomessi.

Gli studenti possono seguire corsi e caricare lavori.

Tutti gli utenti – studenti e professori – possono gestire i loro dati, vedere i corsi e l'insieme di task per ogni corso (ammesso che l'utente sia coinvolto nel corso), e vedere i lavori sottomessi e i punti di valutazione. Ogni studente può vedere solamente i suoi lavori ed i suoi voti. Ogni professore può vedere i lavori assegnati a lui e i voti di questi lavori. L'amministratore del corso, invece, ha diritto di accesso a tutti i dati.

Un corso viene creato o eliminato da un impiegato amministratore. Quando il corso viene creato, almeno un amministratore deve essere inserito. Ulteriori amministratori possono essere aggiunti o eliminati in un secondo momento. L'amministratore può anche cancellare interi corsi.

Le informazioni riguardo gli utenti vengono trasferite automaticamente da un altro sistema, quindi le funzioni che permettono la creazione dei dati degli utenti non sono necessarie.

Tutte le funzioni del sistema possono essere usate solamente dalle persone che sono loggiate.



# Class Diagram – Individuazione delle classi

Ogni **corso** inserito nel sistema ha dei **professori** associati. L'inserimento viene svolto dall'amministratore di un corso, che è anch'esso un professore. Come parte di un corso, i professori possono creare **tasks** e valutare i **lavori sottomessi** dagli **studenti**. Quindi, il professore può assegnare dei punti ad ogni lavoro e fornire un feedback.

L'amministratore del corso definisce quali professori possono valutare quali specifici lavori. Alla fine del corso, l'amministratore del corso provvede al rilascio dei certificati. Il voto di uno studente viene calcolato in base al numero totale di punti per i lavori sottomessi.

Gli studenti possono seguire corsi e caricare lavori.

Tutti gli **utenti** – studenti e professori – possono gestire i loro dati, vedere i corsi e l'insieme di task per ogni corso (ammesso che l'utente sia coinvolto nel corso), e vedere i lavori sottomessi e i punti di valutazione. Ogni studente può vedere solamente i suoi lavori ed i suoi voti. Ogni professore può vedere i lavori assegnati a lui e i voti di questi lavori. L'amministratore del corso, invece, ha diritto di accesso a tutti i dati.

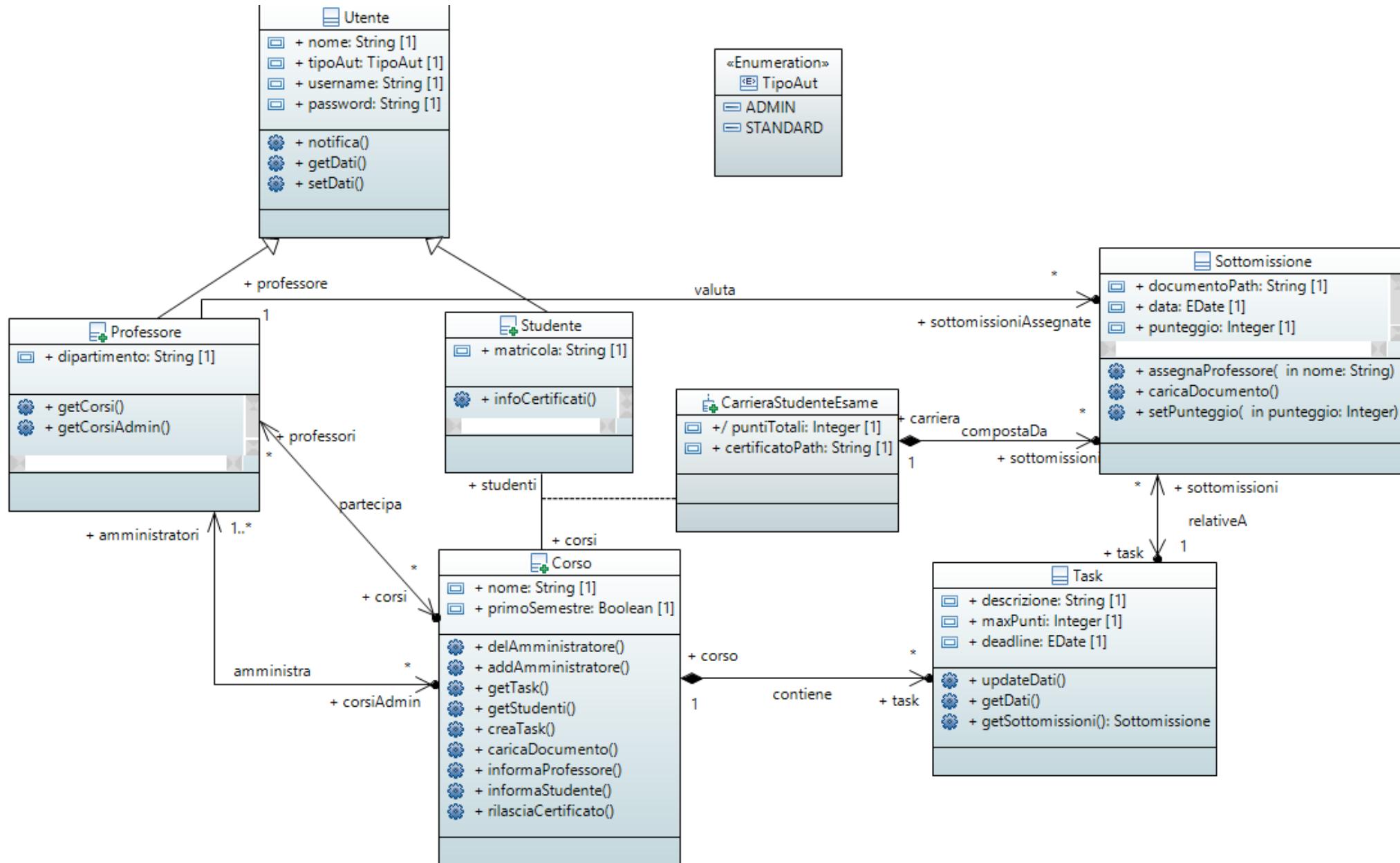
Un corso viene creato o eliminato da un impiegato amministratore. Quando il corso viene creato, almeno un amministratore deve essere inserito. Ulteriori amministratori possono essere aggiunti o eliminati in un secondo momento. L'amministratore può anche cancellare interi corsi.

Le informazioni riguardo gli utenti vengono trasferite automaticamente da un altro sistema, quindi le funzioni che permettono la creazione dei dati degli utenti non sono necessarie.

Tutte le funzioni del sistema possono essere usate solamente dalle persone che sono loggiate.



# Class Diagram – Soluzione



# State Machine Diagram

Si richiede di rappresentare uno state machine diagram per le seguenti situazioni:

**Classe sottomissione:** inizialmente, una sottomissione viene creata. Quando il documento viene caricato, passa in stato uploaded. Quando viene assegnato un professore per la sua correzione passa in stato di assegnato. Quando riceve una valutazione, passa in stato di valutato e viene assegnato un voto. La sottomissione può passare in stato di valutato anche se è stata creata ma nessun documento è stato caricato prima della deadline. In questo caso il voto assegnato è pari a zero.

**Partecipazione di uno studente ad un corso:** per un corso specifico, lo studente si trova nello stato di non valutato, alla prima valutazione passa nello stato parzialmente valutato. In questo stato può ricevere valutazioni intermedie, ed, infine, una valutazione finale. In questo caso, passa nello stato di certificato rilasciato. Un certificato rilasciato può essere positivo e negativo. Questi sottostati possono cambiare se un certificato viene corretto. Se uno studente non ha mai ricevuto alcuna valutazione (ad esempio perché non ha frequentato nessuna lezione e non ha mai completato nessuna delle attività assegnate), rimane nello stato non valutato fino al termine del corso.



# State Machine Diagram

**Classe sottomissione:** inizialmente, una sottomissione viene creata. Quando il documento viene caricato, passa in stato uploaded. Quando viene assegnato un professore per la sua correzione passa in stato di assegnato. Quando riceve una valutazione, passa in stato di valutato e viene assegnato un voto. La sottomissione può passare in stato di valutato anche se è stata creata ma nessun documento è stato caricato prima della deadline. In questo caso il voto assegnato è pari a zero.



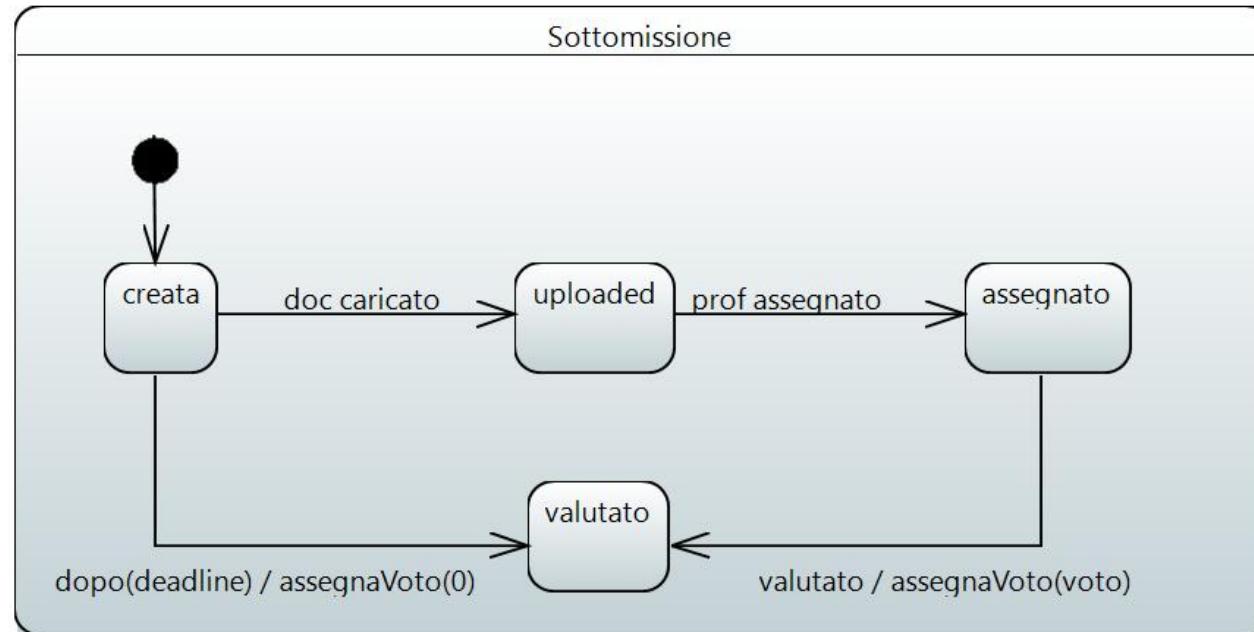
# State Machine Diagram – Stati e Transizioni

**Classe sottomissione:** **inizialmente**, una sottomissione viene **creata**. Quando il documento **viene caricato**, passa in stato **uploaded**. Quando **viene assegnato un professore** per la sua correzione passa in stato di **assegnato**. Quando **riceve una valutazione**, passa in stato di **valutato** e viene assegnato un voto. La sottomissione può passare in stato di valutato anche se è stata creata ma **nessun documento è stato caricato prima della deadline**. In questo caso il voto assegnato è pari a zero.



# State Machine Diagram – Soluzione

**Classe sottomissione:** **inizialmente**, una sottomissione viene **creata**. Quando il documento **viene caricato**, passa in stato **uploaded**. Quando **viene assegnato un professore** per la sua correzione passa in stato di **assegnato**. Quando **riceve una valutazione**, passa in stato di **valutato** e viene assegnato un voto. La sottomissione può passare in stato di valutato anche se è stata creata ma **nessun documento è stato caricato prima della deadline**. In questo caso il voto assegnato è pari a zero.



# State Machine Diagram

**Partecipazione di uno studente ad un corso:** per un corso specifico, lo studente si trova nello stato di non valutato, alla prima valutazione passa nello stato parzialmente valutato. In questo stato può ricevere valutazioni intermedie, ed, infine, una valutazione finale. In questo caso, passa nello stato di certificato rilasciato. Un certificato rilasciato può essere positivo e negativo. Questi sottostati possono cambiare se un certificato viene corretto. Se uno studente non ha mai ricevuto alcuna valutazione (ad esempio perché non ha frequentato nessuna lezione e non ha mai completato nessuna delle attività assegnate), rimane nello stato non valutato fino al termine del corso.

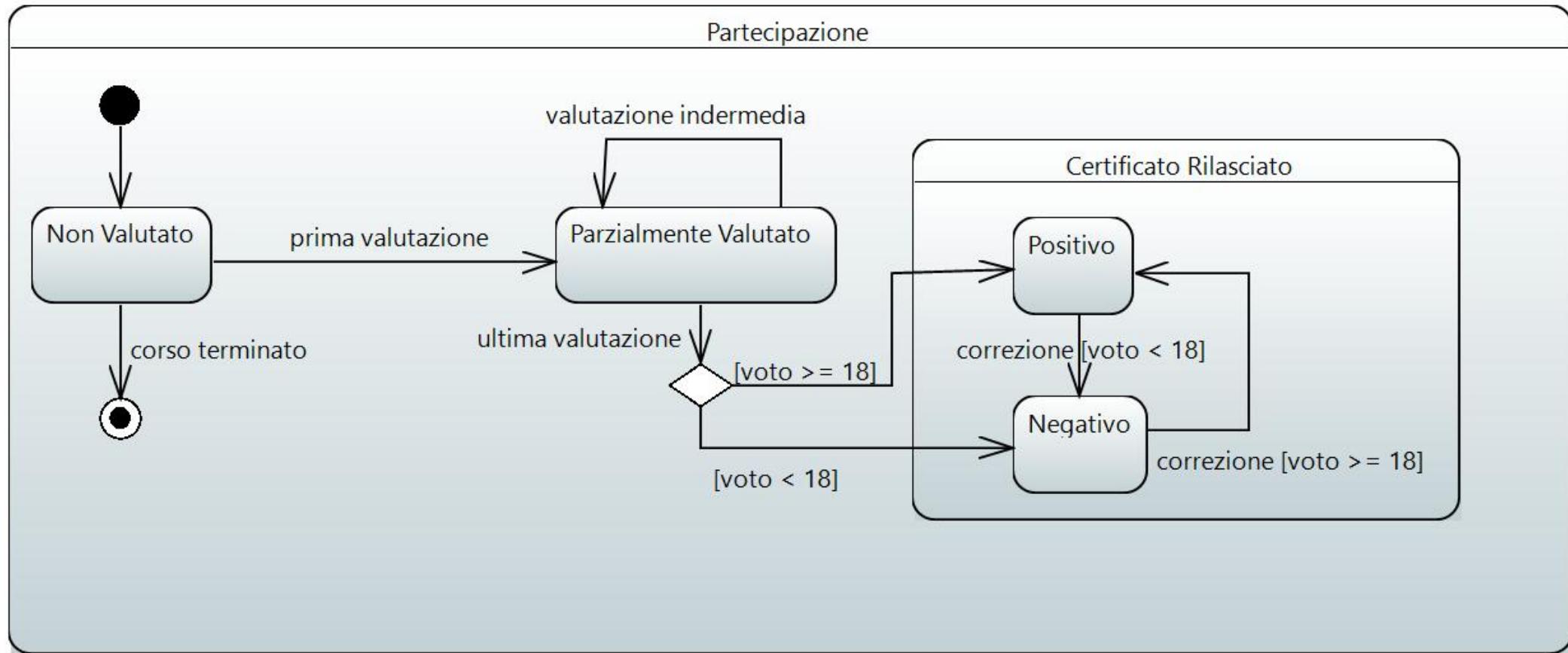


# State Machine Diagram – Stati e Transizioni

**Partecipazione di uno studente ad un corso:** per un corso specifico, lo studente **si trova** nello stato di **non valutato**, alla **prima valutazione** passa nello stato **parzialmente valutato**. In questo stato può **ricevere valutazioni intermedie**, ed, infine, una **valutazione finale**. In questo caso, passa nello stato di **certificato rilasciato**. Un certificato rilasciato può essere **positivo** e **negativo**. Questi sottostati possono cambiare se un certificato viene **corretto**. Se uno studente non ha mai ricevuto alcuna valutazione (ad esempio perché non ha frequentato nessuna lezione e non ha mai completato nessuna delle attività assegnate), **rimane nello stato non valutato fino al termine del corso**.



# State Machine Diagram - Soluzione



# Sequence Diagram

Modellare l'aspetto comunicativo del sistema:

Uno studente carica il compito risolto (documento) all'interno del sistema di sottomissione. Il sistema informa l'amministratore del corso che una nuova sottomissione è presente e conferma allo studente che il documento è correttamente inserita nel sistema.

Nel sequence diagram non mostriamo le azioni necessarie per salvare il documento, dato che non è rilevante nella rappresentazione dello specifico processo di comunicazione.

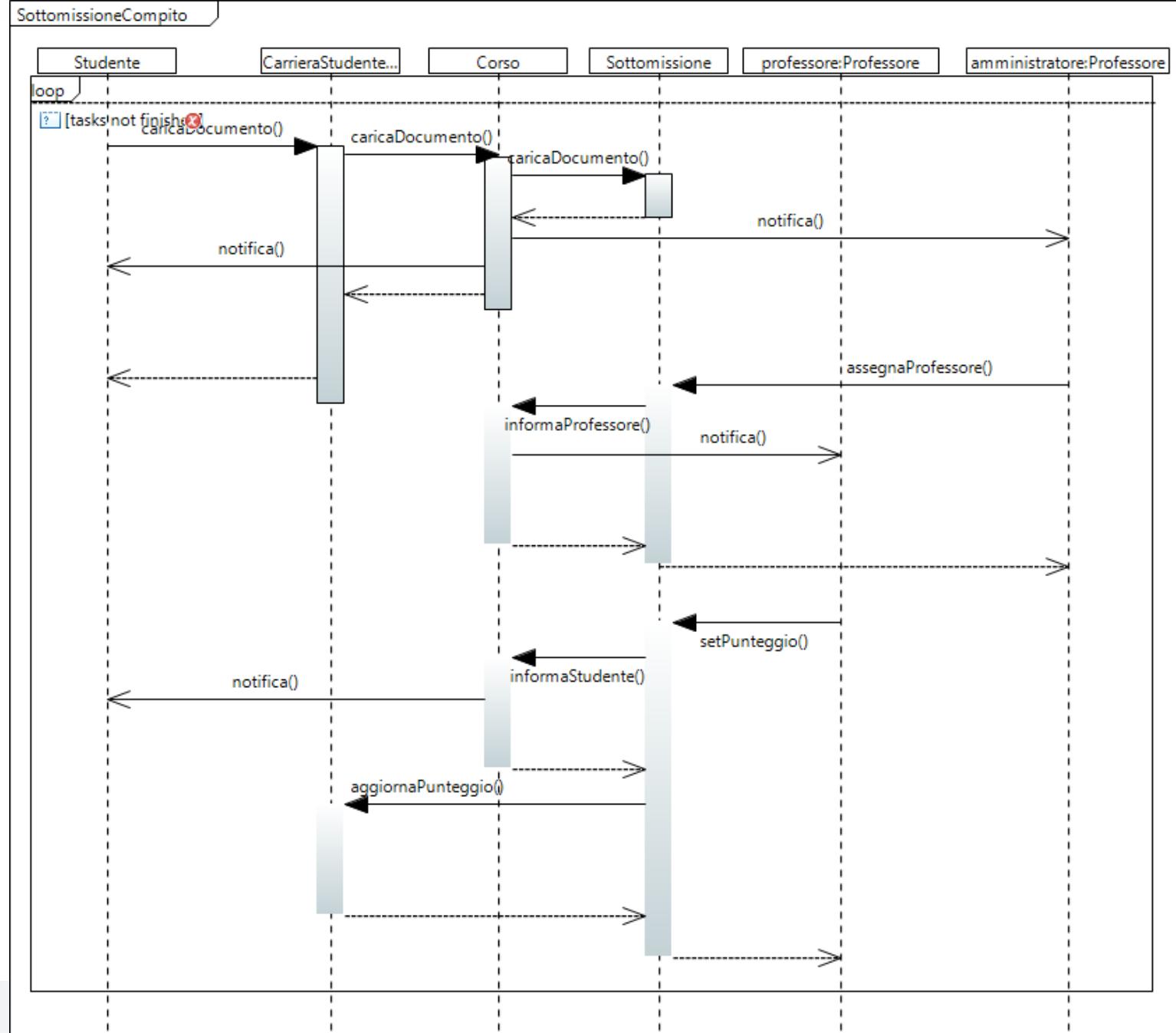
Tramite il sistema di sottomissione, l'amministratore del corso assegna un professore alla correzione della sottomissione. Quando il sistema ha informato il professore che il documento è stato assegnato, il professore valuta la sottomissione. Per fare questo, il professore scarica il documento dal sistema di sottomissione ed inserisce il voto nel sistema.

Quindi, lo studente viene informato che il documento caricato è stato valutato.

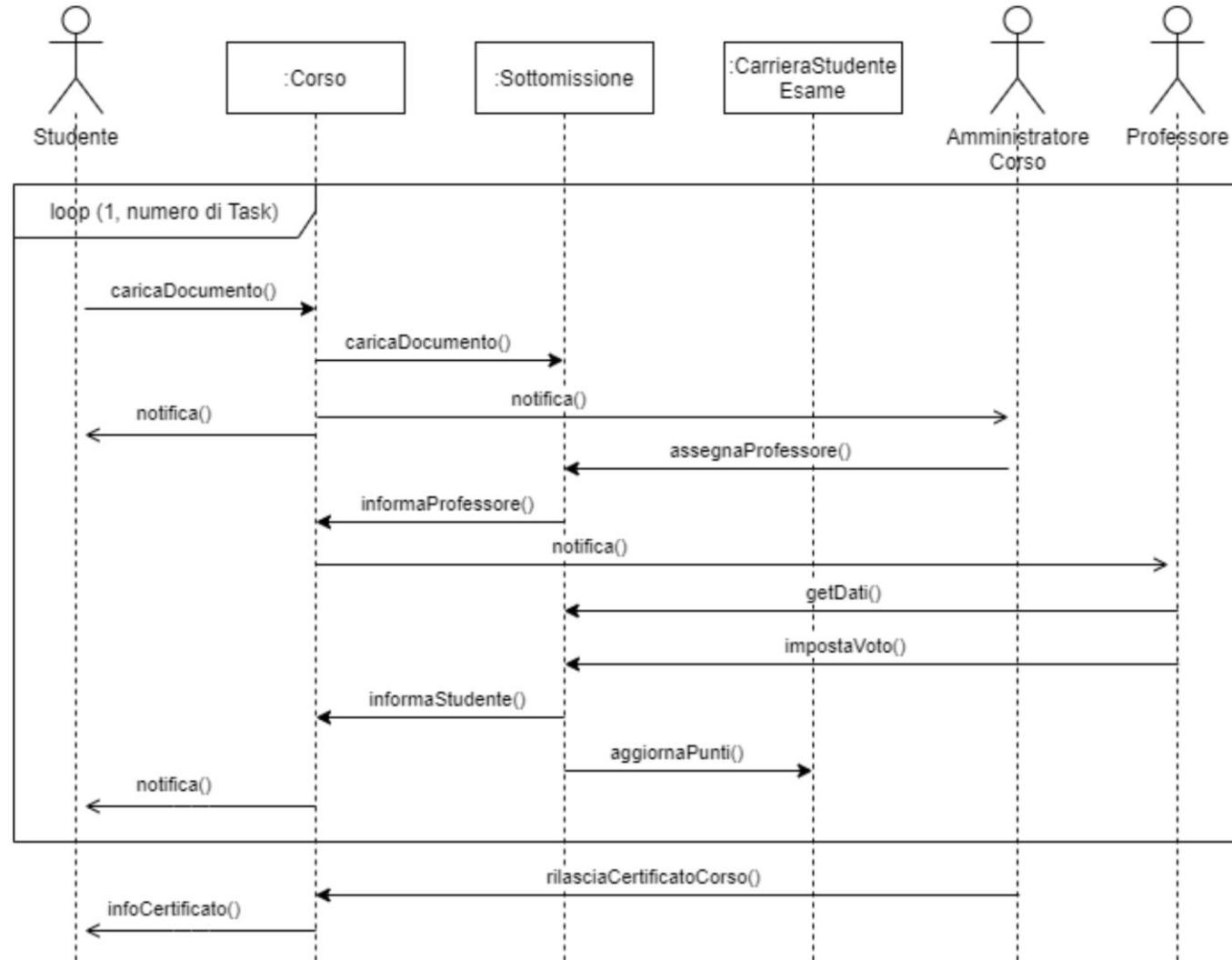
Nota che il flusso di comunicazione descritto non viene eseguito solo una volta ma per ogni task che deve essere completato per un corso.



# Class Diagram - Soluzione 1



# Class Diagram – Soluzione 2



# Activity Diagram

Modellare il rilascio di un certificato:

Assumiamo che sull'interfaccia utente l'amministratore del corso veda una panoramica dei corsi assegnati. Inizialmente l'amministratore del corso seleziona il corso per cui vuole rilasciare i certificati. Gli studenti iscritti al corso vengono quindi mostrati. L'amministratore del corso può selezionare se rilasciare i certificati per tutti o solo per determinati studenti. Nel secondo caso, l'amministratore deve anche specificare per quali studenti deve essere rilasciato il certificato. Quindi, i voti vengono calcolati dal sistema e inviati alla segreteria, che rilascia i certificati e informa gli studenti del voto preso.



# Activity Diagram – Individuazione delle attività

Assumiamo che sull'interfaccia utente l'amministratore del corso veda una panoramica dei corsi assegnati. Inizialmente l'amministratore del corso **seleziona il corso** per cui vuole rilasciare i certificati. Gli studenti iscritti al corso **vengono quindi mostrati**. L'amministratore del corso può selezionare se **rilasciare i certificati** per tutti o solo per determinati studenti. Nel secondo caso, l'amministratore deve anche **specificare per quali studenti** deve essere rilasciato il certificato. Quindi, i **voti vengono calcolati** dal sistema e **inviati** alla segreteria, che **rilascia i certificati** e **informa gli studenti** del voto preso.

In *italico* gli attori che svolgono le attività, nota che dove non indicato viene svolto dal *sistema*.

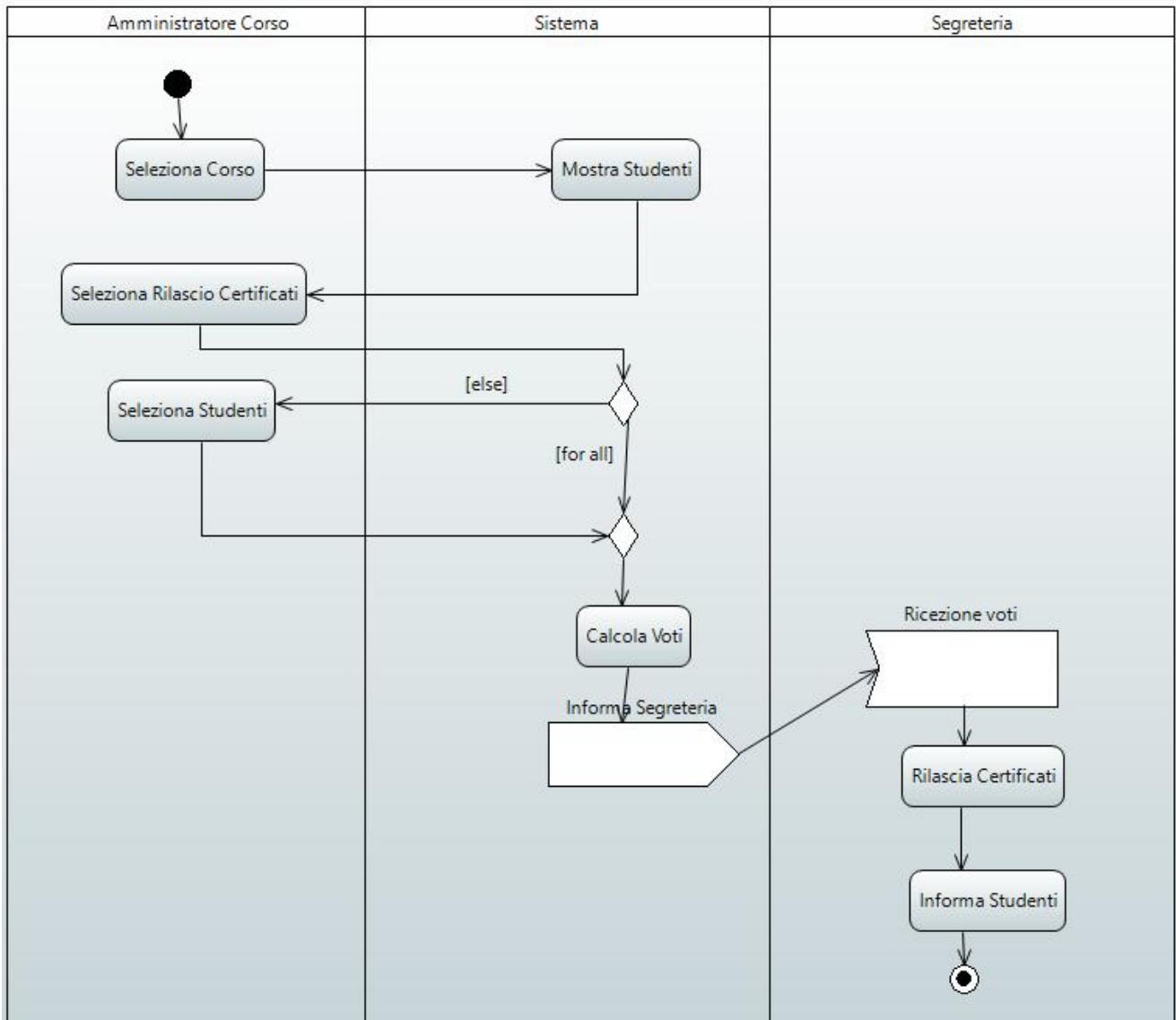


# Activity Diagram - Soluzione

Assumiamo che sull'interfaccia utente l'amministratore del corso veda una panoramica dei corsi assegnati.

Inizialmente l'amministratore del corso **seleziona il corso** per cui vuole rilasciare i certificati. Gli studenti iscritti al corso **vengono quindi mostrati**. L'amministratore del corso può selezionare se **rilasciare i certificati** per tutti o solo per determinati studenti. Nel secondo caso, l'amministratore deve anche **specificare per quali studenti** deve essere rilasciato il certificato. Quindi, i **voti vengono calcolati** dal sistema e **inviati alla segreteria**, che **rilascia i certificati e informa gli studenti** del voto preso.

RilascioCertificati



# Diagrammi Papyrus disponibili qui:

<https://github.com/nicopellegrinelli/EsercitazionildS-public/tree/solution>



UNIVERSITÀ  
DEGLI STUDI  
DI BERGAMO

Dipartimento  
di Ingegneria Gestionale,  
dell'Informazione e della Produzione