

Università degli Studi di Napoli Federico II

**Corso di Laurea in Ingegneria Informatica**

**Corso di Ingegneria del Software**

**Prof. A.R. Fasolino - A.A. 2024 - 25**

Immagine che contiene calzature, cartone animato, vestiti, clipart

Descrizione generata automaticamente

***TEMPLATE PER IL PROGETTO   
DI INGEGNERIA DEL SOFTWARE***

***“SISTEMA GESTIONALE DI TASK DIDATTICI A PUNTI”***

**INDICE**

[1. Specifiche informali 1](#_Toc200110538)

[2. Analisi e specifica dei requisiti 2](#_Toc200110539)

[2.1 Analisi nomi-verbi 2](#_Toc200110540)

[2.2 Revisione dei requisiti 3](#_Toc200110541)

[2.3 Glossario dei termini 4](#_Toc200110542)

[2.4 Classificazione dei requisiti 4](#_Toc200110543)

[2.4.1 Requisiti funzionali 4](#_Toc200110544)

[2.4.2 Requisiti sui dati 6](#_Toc200110545)

[2.4.3 Vincoli / Altri requisiti 6](#_Toc200110546)

[2.5 Modellazione dei casi d’uso 7](#_Toc200110547)

[2.5.1 Attori e casi d’uso 7](#_Toc200110548)

[2.5.2 Diagramma dei casi d’uso 8](#_Toc200110549)

[2.5.3 Scenari 8](#_Toc200110550)

[2.6 Diagramma delle classi 10](#_Toc200110551)

[2.7 Diagrammi di sequenza 12](#_Toc200110552)

[2.7.1 AssegnaTaskAClasse 12](#_Toc200110553)

[2.7.2 ConsegnaSoluzione 12](#_Toc200110554)

[2.7.3 ValutaConsegna 13](#_Toc200110555)

[2.8 Diagramma delle classi raffinato 13](#_Toc200110556)

[3. Piano di test funzionale 14](#_Toc200110557)

[3.1 AssegnaTaskAClasse 14](#_Toc200110558)

[3.2 ConsegnaSoluzione 15](#_Toc200110559)

[3.3 ValutaConsegna 16](#_Toc200110560)

[4. Progettazione 17](#_Toc200110561)

[4.1 Diagramma delle classi 17](#_Toc200110562)

[4.1.1 Pattern BCED 17](#_Toc200110563)

[4.1.1.1 Package Boundary 17](#_Toc200110564)

[4.1.1.2 Package Controller 17](#_Toc200110565)

[4.1.1.3 Package Entity 18](#_Toc200110566)

[4.1.1.4 Modello E/R 18](#_Toc200110567)

[4.1.1.5 Package Database 19](#_Toc200110568)

[4.2 Diagrammi di sequenza 20](#_Toc200110569)

[4.2.1 AssegnaTaskAClasse 21](#_Toc200110570)

[1.1.1 ConsegnaSoluzione 22](#_Toc200110571)

[4.2.2 ValutaConsegna 23](#_Toc200110572)

[5. Implementazione 24](#_Toc200110573)

[5.1 Package Database 24](#_Toc200110574)

[5.2 Package Entity 24](#_Toc200110575)

[5.3 Package Controller 25](#_Toc200110576)

[5.4 Package Boundary 25](#_Toc200110577)

[5.5 Package DTO 25](#_Toc200110578)

[5.6 Dipendenze: Jakarta 25](#_Toc200110579)

[5.7 Diagramma di Deployment 26](#_Toc200110580)

[6. Testing 26](#_Toc200110581)

[6.1 Test funzionale 27](#_Toc200110582)

# Specifiche informali

**Scenario:**

*Si desidera sviluppare un sistema software per la gestione di task didattici assegnati da docenti a studenti, con un meccanismo di punteggio che consente di generare classifiche dinamiche basate sui risultati ottenuti dagli studenti.*

**Descrizione del Sistema:**

*Il sistema consente l’accesso di docenti e studenti a una piattaforma digitale mediante autenticazione tramite credenziali personali. Al momento della registrazione, ciascun utente deve specificare il proprio ruolo (studente o docente), oltre a fornire nome, cognome e indirizzo e-mail istituzionale. Ogni docente ha la possibilità di creare una o più classi virtuali, ciascuna identificata da un nome e da un codice univoco. L’iscrizione di uno studente a una classe può avvenire in due modalità: il docente può iscrivere direttamente lo studente alla propria classe, selezionandolo da un elenco di studenti iscritti alla piattaforma, oppure lo studente stesso può iscriversi autonomamente, inserendo il codice univoco della classe fornito dal docente. Ogni studente può appartenere ad una sola classe (per semplicità).*

*All’interno di ogni classe, il docente può creare e assegnare task didattici agli studenti. Ogni task è definito da un titolo, una descrizione, una data di scadenza, un numero massimo di punti assegnabili. Gli studenti possono visualizzare l’elenco dei task assegnati direttamente all’interno del proprio profilo personale. Una volta selezionato un task, possono procedere con la consegna caricando la propria soluzione, che dovrà essere fornita in formato testuale attraverso un’apposita interfaccia. Il docente ha la facoltà di valutare ogni consegna assegnando un punteggio compreso tra zero e il massimo indicato nel task.*

*Il sistema mantiene lo storico di tutti i task svolti da ciascuno studente e i relativi punteggi ottenuti.*

*Ogni studente può visualizzare la classifica relativa esclusivamente alla propria classe, attraverso un’apposita sezione del profilo. Le classifiche disponibili sono due: una basata sul punteggio totale accumulato e una basata sul numero di task completati. Queste classifiche consentono agli studenti di confrontare le proprie performance con quelle degli altri membri della stessa classe, incentivando la partecipazione attiva alle attività didattiche. Per semplicità si può supporre che la classifica mostri solo un numero limitato di studente (ex. i primi tre, o i primi cinque)*

*Ogni studente ha accesso a un profilo personale dove può consultare il totale dei task assegnati, il punteggio totale ottenuto, e la media dei voti ottenuti. I docenti, attraverso un’interfaccia dedicata, possono visualizzare l’insieme delle classi create e, per ciascuna classe, visualizzare l’elenco degli studenti iscritti e l’elenco dei task assegnati.*

*Il sistema dovrà essere accessibile via web sia da desktop che da dispositivi mobili, con un’interfaccia grafica intuitiva e responsiva. È previsto un sistema di notifiche che avvisi gli studenti della pubblicazione di nuovi task e dell’avvenuta correzione delle proprie consegne.*

# Analisi e specifica dei requisiti

## Analisi nomi-verbi

*Il sistema consente l’accesso di docenti e studenti a una piattaforma digitale mediante autenticazione tramite credenziali personali. Al momento della registrazione, ciascun utente deve specificare il proprio ruolo (studente o docente), oltre a fornire nome, cognome e indirizzo e-mail istituzionale. Ogni docente ha la possibilità di creare una o più classi virtuali, ciascuna identificata da un nome e da un codice univoco. L’iscrizione di uno studente a una classe può avvenire in due modalità: il docente può iscrivere direttamente lo studente alla propria classe, selezionandolo da un elenco di studenti iscritti alla piattaforma, oppure lo studente stesso può iscriversi autonomamente, inserendo il codice univoco della classe fornito dal docente. Ogni studente può appartenere ad una sola classe (per semplicità).*

*All’interno di ogni classe, il docente può creare e assegnare task didattici agli studenti. Ogni task è definito da un titolo, una descrizione, una data di scadenza, un numero massimo di punti assegnabili. Gli studenti possono visualizzare l’elenco dei task assegnati direttamente all’interno del proprio profilo personale. Una volta selezionato un task, possono procedere con la consegna caricando la propria soluzione, che dovrà essere fornita in formato testuale attraverso un’apposita interfaccia. Il docente ha la facoltà di valutare ogni consegna assegnando un punteggio compreso tra zero e il massimo indicato nel task.*

*Il sistema mantiene lo storico di tutti i task svolti da ciascuno studente e i relativi punteggi ottenuti.*

*Ogni studente può visualizzare la classifica relativa esclusivamente alla propria classe, attraverso un’apposita sezione del profilo. Le classifiche disponibili sono due: una basata sul punteggio totale accumulato e una basata sul numero di task completati. Queste classifiche consentono agli studenti di confrontare le proprie performance con quelle degli altri membri della stessa classe, incentivando la partecipazione attiva alle attività didattiche. Per semplicità si può supporre che la classifica mostri solo un numero limitato di studente (ex. i primi tre, o i primi cinque)*

*Ogni studente ha accesso a un profilo personale dove può consultare il totale dei task assegnati, il punteggio totale ottenuto, e la media dei voti ottenuti. I docenti, attraverso un’interfaccia dedicata, possono visualizzare l’insieme delle classi create e, per ciascuna classe, visualizzare l’elenco degli studenti iscritti e l’elenco dei task assegnati.*

*Il sistema dovrà essere accessibile via web sia da desktop che da dispositivi mobili, con un’interfaccia grafica intuitiva e responsiva. È previsto un sistema di notifiche che avvisi gli studenti della pubblicazione di nuovi task e dell’avvenuta correzione delle proprie consegne.*

*LEGENDA:  
Classe  
Attributo  
Funzionalità  
Attore*

*Classe-Attore*

## Revisione dei requisiti

1. Il sistema deve offrire all’Utente una funzionalità per registrarsi.
2. Il sistema deve offrire all’Utente una funzionalità per accedere.
3. Di ogni Utente si vuole memorizzare nome, cognome, e-mail istituzionale e password.
4. Il sistema deve permettere ai docenti di creare una o più classi virtuali.
5. Il sistema deve permettere ai docenti di visualizzare l’insieme delle classi create.
6. Il sistema deve permettere ai docenti di visualizzare l’elenco degli studenti iscritti ad una classe creata.
7. Il sistema deve permettere ai docenti di visualizzare l’elenco dei task assegnati in una classe creata.
8. Di ogni classe virtuale si vuole memorizzare nome e codice univoco.
9. Il sistema deve permettere a uno studente di iscriversi ad una classe virtuale, specificando il codice univoco della classe.
10. Uno studente può iscriversi ad una sola classe virtuale.
11. Il sistema deve permettere al docente di iscrivere uno studente ad una propria classe.
12. Il sistema deve consentire ai docenti di creare e assegnare task didattici all’interno delle classi virtuali.
13. Di ogni task didattico si vuole memorizzare titolo, descrizione, data di scadenza e numero massimo di punti assegnabili.
14. Il sistema deve permettere agli studenti di visualizzare i task assegnati all'interno del proprio profilo personale.
15. Il sistema deve consentire agli studenti di selezionare un task e caricare una propria soluzione in formato testuale, tramite un’apposita interfaccia.
16. Il sistema deve consentire al docente di assegnare un punteggio alla consegna dello studente.
17. Il sistema deve mantenere uno storico dei task svolti da ciascuno studente e i relativi punteggi ottenuti.
18. Il sistema deve generare una classifica per ogni classe virtuale.
19. Il sistema deve consentire allo studente di visualizzare la classifica relativa esclusivamente alla propria classe.
20. Il sistema deve offrire due tipologie di classifica, una basata sul punteggio totale accumulato e una basata sul numero di task completati.
21. Solo i primi cinque studenti nella classifica sono visualizzati.
22. Il sistema deve fornire allo studente l’accesso ad un profilo personale.
23. Di ogni profilo personale si vuole memorizzare il totale dei task assegnati, il punteggio totale ottenuto e la media dei voti ottenuti.
24. Il sistema deve avere un’interfaccia grafica intuitiva e responsiva.
25. Il sistema deve notificare gli studenti della pubblicazione di nuovi task e dell’avvenuta correzione delle proprie consegne.
26. Il punteggio assegnato dal docente alla consegna deve essere compreso tra 0 e il valore massimo specificato nel task.
27. Il sistema deve permettere ai docenti di visualizzare l’elenco degli studenti iscritti alla piattaforma.
28. Il sistema deve permettere ai docenti di visualizzare l’elenco degli studenti iscritti ad una sua classe.

## Glossario dei termini

|  |  |  |
| --- | --- | --- |
| **Termine** | **Descrizione** | **Sinonimi** |
| **Utente** | Persona che utilizza la piattaforma, può essere uno studente o un docente |  |
| **Studente** | Utente con il ruolo di apprendente, iscritto a una classe virtuale e destinatario dei task |  |
| **Docente** | Utente con il ruolo di insegnante, creatore delle classi virtuali e assegnatore dei task |  |
| **Task didattico** | Attività didattica assegnata agli studenti all’interno di una classe | Task |
| **Consegna** | Risultato del lavoro svolto da uno studente su un task, inviato tramite piattaforma |  |
| **E-mail istituzionale** | Indirizzo di posta elettronica univoco, rilasciato degli istituiti di formazione verso i propri studenti |  |

## 

## Classificazione dei requisiti

### Requisiti funzionali

|  |  |  |
| --- | --- | --- |
| **ID** | **Requisito** | **Origine (n. frase dei requisiti revisionati)** |
| **RF01** | Il sistema deve offrire all’Utente una funzionalità per registrarsi | 1 |
| **RF02** | Il sistema deve offrire all’Utente una funzionalità per accedere | 2 |
| **RF03** | Il sistema deve permettere ai docenti di creare una o più classi virtuali | 4 |
| **RF04** | Il sistema deve permettere ai docenti di visualizzare l’insieme delle classi create | 5 |
| **RF05** | Il sistema deve permettere ai docenti di visualizzare l’elenco degli studenti iscritti ad una classe creata | 6 |
| **RF06** | Il sistema deve permettere ai docenti di visualizzare l’elenco dei task assegnati in una classe creata | 7 |
| **RF07** | Il sistema deve permettere a uno studente di iscriversi ad una classe virtuale, specificando il codice univoco della classe | 9 |
| **RF08** | Il sistema deve permettere al docente di iscrivere uno studente ad una propria classe, selezionandolo dall’elenco degli studenti iscritti alla piattaforma | 11 |
| **RF09** | Il sistema deve consentire ai docenti di creare e assegnare task didattici all’interno delle classi virtuali | 12 |
| **RF10** | Il sistema deve permettere agli studenti di visualizzare i task assegnati all'interno del proprio profilo personale | 14 |
| **RF11** | Il sistema deve consentire agli studenti di selezionare un task e caricare una propria soluzione in formato testuale, tramite un’apposita interfaccia | 15 |
| **RF12** | Il sistema deve consentire al docente di assegnare un punteggio alla consegna dello studente. Il punteggio deve essere compreso tra 0 e il valore massimo specificato nel task | 16 |
| **RF13** | Il sistema deve mantenere uno storico dei task svolti da ciascuno studente e i relativi punteggi ottenuti | 17 |
| **RF14** | Il sistema deve generare una classifica per ogni classe virtuale | 18 |
| **RF15** | Il sistema deve consentire allo studente di visualizzare la classifica relativa esclusivamente alla propria classe | 19 |
| **RF16** | Il sistema deve offrire due tipologie di classifica, una basata sul punteggio totale accumulato e una basata sul numero di task completati | 20 |
| **RF17** | Il sistema deve fornire allo studente l’accesso ad un profilo personale | 22 |
| **RF18** | Il sistema deve notificare gli studenti della pubblicazione di nuovi task e dell’avvenuta correzione delle proprie consegne | 25 |
| **RF19** | Il sistema deve permettere ai docenti di visualizzare l’elenco degli studenti iscritti alla piattaforma | 27 |
| **RF20** | Il sistema deve permettere ai docenti di visualizzare l’elenco degli studenti iscritti ad una propria classe | 28 |

### Requisiti sui dati

|  |  |  |
| --- | --- | --- |
| **ID** | **Requisito** | **Origine (n. frase dei requisiti revisionati)** |
| **RD01** | Di ogni Utente si vuole memorizzare nome, cognome, e-mail istituzionale e password | 3 |
| **RD02** | Di ogni classe virtuale si vuole memorizzare nome e codice univoco | 8 |
| **RD03** | Di ogni task didattico si vuole memorizzare titolo, descrizione, data di scadenza e numero massimo di punti assegnabili | 13 |
| **RD04** | Di ogni profilo personale si vuole memorizzare il totale dei task assegnati, il punteggio totale ottenuto e la media dei voti ottenuti | 23 |

### Vincoli / Altri requisiti

|  |  |  |
| --- | --- | --- |
| **ID** | **Requisito** | **Origine (n. frase dei requisiti revisionati)** |
| **V01** | Uno studente può iscriversi ad una sola classe virtuale | 10 |
| **V02** | Solo i primi cinque studenti nella classifica sono visualizzati | 21 |
| **V03** | Il sistema deve avere un’interfaccia grafica intuitiva e responsiva | 24 |
| **V04** | Il punteggio assegnato dal docente alla consegna deve essere compreso tra 0 e il valore massimo specificato nel task | 26 |

## Modellazione dei casi d’uso

### Attori e casi d’uso

***Attori Primari:***

* Utente

***Attori Secondari:***

* Sistema Notifiche
* Utente Registrato
* Studente Iscritto
* Studente Non Iscritto
* Docente

***Casi d’uso:***

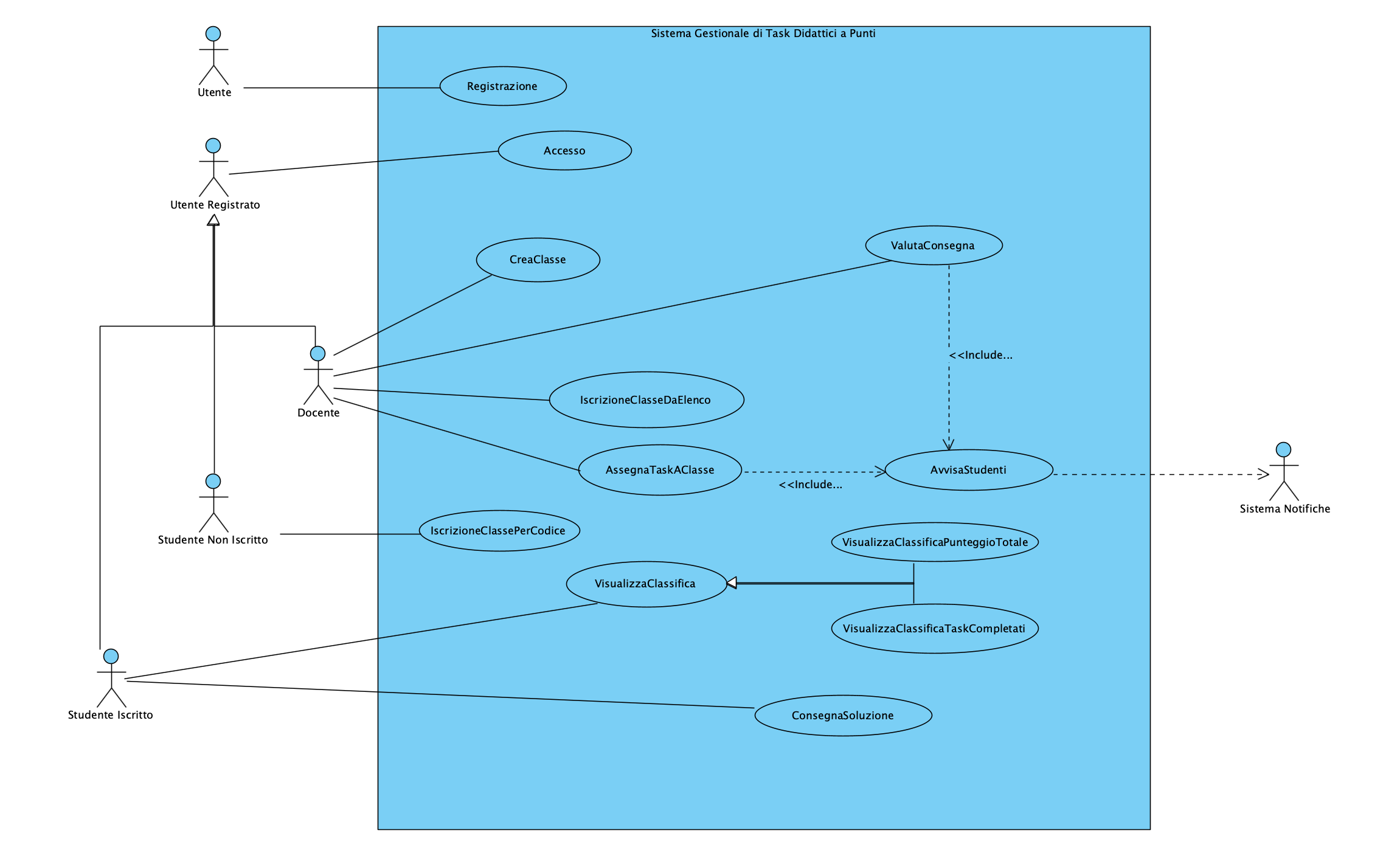
* **UC1**: Registrazione
* **UC2**: Accesso
* **UC3**: CreaClasse
* **UC4**: IscrizioneClassePerCodice
* **UC5**: IscrizioneClasseDaElenco
* **UC6**: AssegnaTask
* **UC7**: ConsegnaSoluzione
* **UC8**: ValutaConsegna
* **UC9**: VisualizzaClassifica
* **UC10**: VisualizzaClassificaPunteggioTotale
* **UC11**: VisualizzaClassificaTaskCompletati

***Casi d’uso di inclusione:***

* **UC12**: AvvisaStudenti

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Caso d’uso | Attori Primari | Attori Secondari | Incl. / Ext. | Requisiti corrispondenti |
| UC1: Registrazione | Utente | - | - | RF01 |
| UC2: Accesso | Utente Registrato | - | - | RF02, RF17 |
| UC3: CreaClasse | Docente | - | - | RF03 |
| UC4: IscrizioneClassePerCodice | Studente Non Iscritto | - | - | RF07 |
| UC5: IscrizioneClasseDaElenco | Docente | - | - | RF08, RF19 |
| UC6: AssegnaTaskAClasse | Docente | - | Include AvvisaStudenti | RF04, RF09 |
| UC7: ConsegnaSoluzione | Studente Iscritto | - | - | RF10, RF11 |
| UC8: ValutaConsegna | Docente | - | Include AvvisaStudenti | RF06, RF12, RF20 |
| UC9: VisualizzaClassifica | Studente Iscritto | - | - | RF13, RF14, RF15, RF16 |
| UC10: VisualizzaClassificaPunteggioTotale | Studente Iscritto | - | - | RF13 RF14, RF15, |
| UC11: VisualizzaClassificaTaskCompletati | Studente Iscritto | - | - | RF13 RF14, RF15, |
| UC12: AvvisaStudenti | - | Sistema Notifiche | È incluso in ValutaConsegna e AssegnaTask | RF18 |

### Diagramma dei casi d’uso



### Scenari

|  |  |
| --- | --- |
| Caso d’uso: | AssegnaTaskAClasse |
| Attore primario | Docente |
| Attore secondario | Sistema Notifiche |
| Descrizione | Un docente vuole assegnare un task ad una sua classe, selezionata dall’elenco delle classi da lui create |
| Pre-Condizioni | Il Docente ha effettuato l’accesso |
| Sequenza di eventi  principale | 1. Il caso d’suo inizia quando il Docente seleziona l’elenco delle classi create 2. Il sistema mostra le classi create dal Docente 3. if ci sono classi create    1. Il Docente seleziona dall’elenco una classe    2. Il sistema mostra una serie di opzioni    3. Il Docente seleziona l’opzione per aggiungere un nuovo task    4. Il Docente specifica titolo, descrizione, data di scadenza e max punti assegnabili    5. if inserimento task andato a buon fine   3.5.1 <<include>> AvvisaStudenti |
| Post-Condizioni | Il Docente ha assegnato un nuovo task ad una sua classe ed esso sarà visibile dall’elenco dei task assegnati |
| Casi d’uso correlati | *AvvisaStudenti* |
| Sequenza di eventi  alternativi | Al punto 3.1, se non ci sono classi create, il sistema restituisce un messaggio di avviso.  Al punto 3.5, se l’inserimento del task non è andato a buon fine, il sistema restituisce un messaggio di errore. |

|  |  |
| --- | --- |
| Caso d’uso: | ConsegnaSoluzione |
| Attore primario | Studente Iscritto |
| Attore secondario | - |
| Descrizione | Uno Studente Iscritto, dopo aver visualizzato i Task all’interno del proprio profilo personale, ne seleziona uno e carica la propria soluzione |
| Pre-Condizioni | Lo Studente Iscritto ha effettuato l’accesso |
| Sequenza di eventi  principale | 1. Il caso d’uso inizia quando lo Studente Iscritto accede all’elenco dei Task assegnati, all’interno del proprio profilo personale 2. Il sistema mostra i Task assegnati 3. **if** ci sono Task assegnati    1. Lo Studente Iscritto seleziona un Task    2. Il sistema indirizza lo Studente Iscritto nella sezione per la consegna della soluzione    3. Lo Studente Iscritto inserisce la propria soluzione    4. Lo Studente Iscritto consegna la propria soluzione    5. Il sistema notifica allo Studente Iscritto che la consegna è avvenuta con successo |
| Post-Condizioni | Lo Studente Iscritto ha consegnato la propria soluzione per il Task scelto e il sistema aggiorna le statistiche dello Studente Iscritto sui Task completati |
| Casi d’uso correlati | - |
| Sequenza di eventi  alternativi | Al punto 3, se non ci sono Task disponibili, il sistema restituisce un messaggio di errore |

|  |  |
| --- | --- |
| Caso d’uso: | ValutaConsegna |
| Attore primario | Docente |
| Attore secondario | - |
| Descrizione | Il Docente valuta la Consegna di uno Studente relativa ad un determinato Task |
| Pre-Condizioni | Il Docente ha effettuato l’accesso al sistema  Il Docente ha selezionato una Classe Virtuale tra quelle da lui create |
| Sequenza di eventi  principale | 1. Il Docente seleziona uno specifico Task della Classe 2. Il sistema restituisce un elenco di Consegne dei vari Studenti 3. **if** Ci sono Consegne    1. Il Docente seleziona dall’elenco una Consegna specifica    2. Il sistema mostra la Consegna selezionata    3. **if** la Consegna non è stata ancora valutata       1. Il Docente assegna un punteggio alla Consegna       2. Il sistema aggiorna le statistiche dello Studente       3. *<<include>> AvvisaStudenti*    4. **else**       1. Il sistema avverte il Docente che la consegna è già stata valutata |
| Post-Condizioni | Il punteggio assegnato alla Consegna viene salvato |
| Casi d’uso correlati | *AvvisaStudenti* |
| Sequenza di eventi  alternativi | Al punto 3, se non ci sono Consegne per il Task selezionato, il sistema mostra un messaggio di mancanza di Consegne  Al punto 3.3.1, se il punteggio assegnato non è valido, il sistema mostra un avviso al Docente, facendogli re-inserire il punteggio, ritornando al punto 3.3 |

## Diagramma delle classi

Immagine che contiene diagramma, schermata, linea, testo

Descrizione generata automaticamente

|  |  |
| --- | --- |
| **RESPONSABILITÀ** | **CLASSE** |
| *Registrazione* | **Piattaforma** |
| *Accesso* | **Piattaforma** |
| *Creazione Classe* | **Docente** |
| *Visualizza Classi Create* | **Docente** |
| *Visualizza Studenti Iscritti* | **ClasseVirtuale** |
| *Visualizza Task Assegnati* | **ClasseVirtuale** |
| *Iscrizione a Classe* | **ClasseVirtuale** |
| *Creazione Task* | **ClasseVirtuale** |
| *Visualizza Classifiche* | **ClasseVirtuale** |
| *Valuta Consegna* | **Task** |
| *Crea Consegna* | **Task** |
| *Visualizza Statistiche Personali* | **Studente** |
| *Consegna Soluzione* | **Consegna** |
| *Aggiorna NumTaskCompletati* | **Consegna** |
| *Aggiorna PunteggioTotaleOttenuto* | **Consegna** |

*Registrazione* e *Accesso* sono responsabilità di **Piattaforma**, in quanto <<information expert>> di Utente.

*Creazione Classe* e *Visualizza Classi Create* sono responsabilità di **Docente**, in quanto <<information expert>> di ClasseVirtuale.

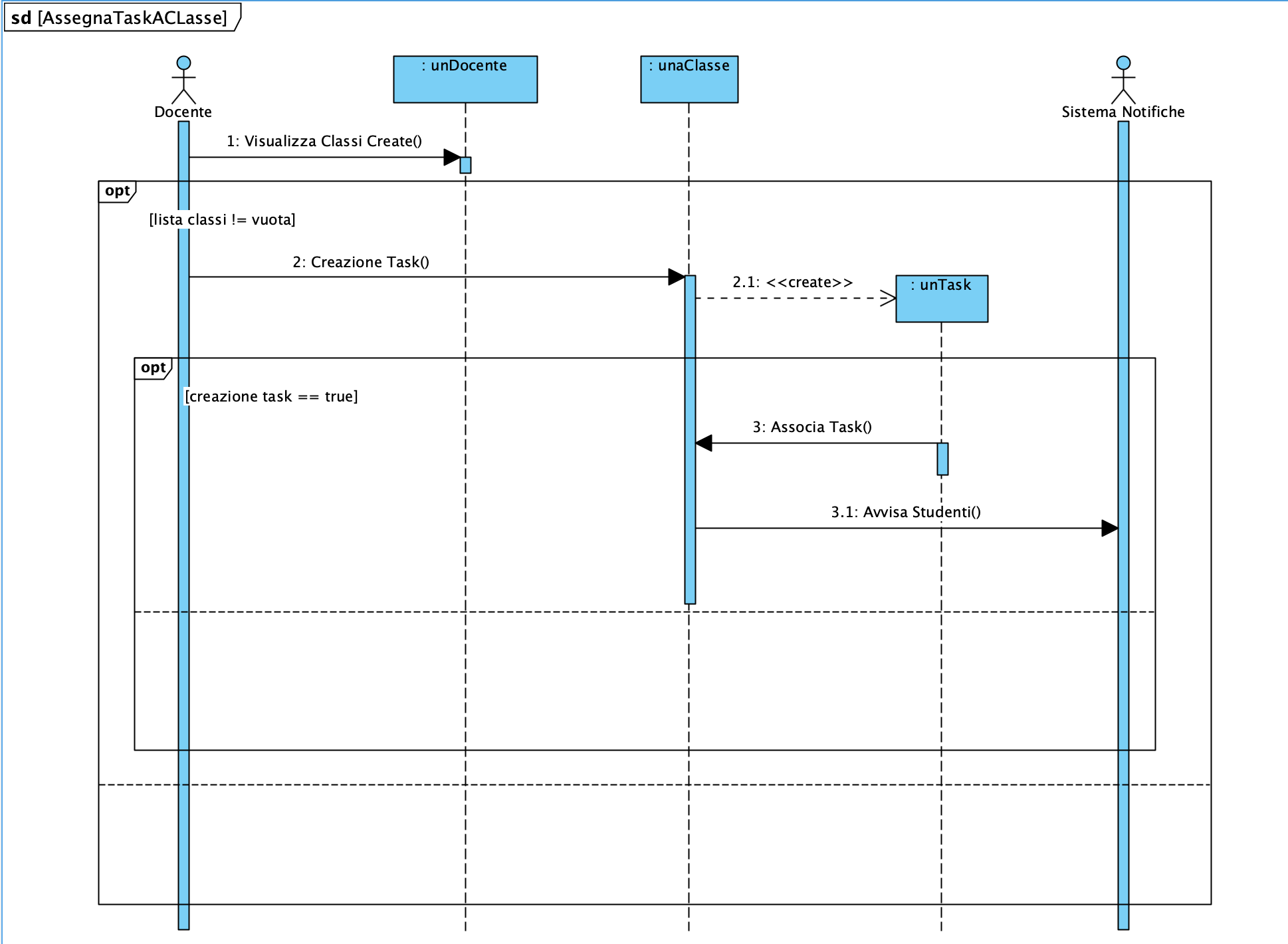
*Visualizza Studenti Iscritti*, *Visualizza Task Assegnati*, *Iscrizione a Classe*, *Creazione Task e* *Visualizza Classifiche* sono responsabilità di **ClasseVirtuale**, in quanto <<information expert>> di Studenti e Task.

*Valuta Consegna* e *Consegna Soluzione* sono responsabilità di **Task**, in quanto <<information expert>> di Consegna.

*Aggiorna NumTaskCompletati* e *Aggiorna PunteggioTotaleOttenuto* sono responsabilità di **Consegna**, per il pattern <<low coupling>> con Studente.

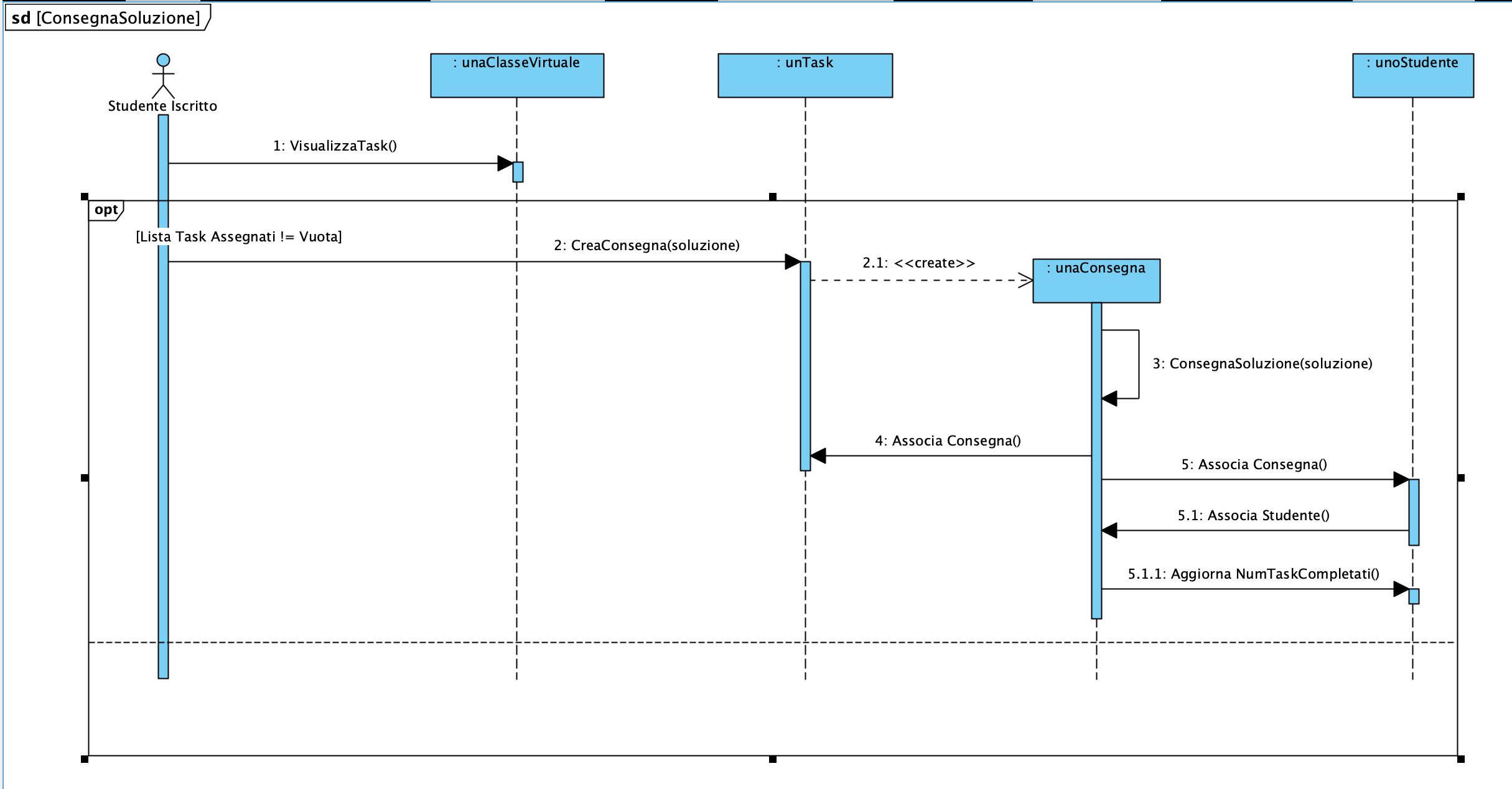
## Diagrammi di sequenza

### AssegnaTaskAClasse



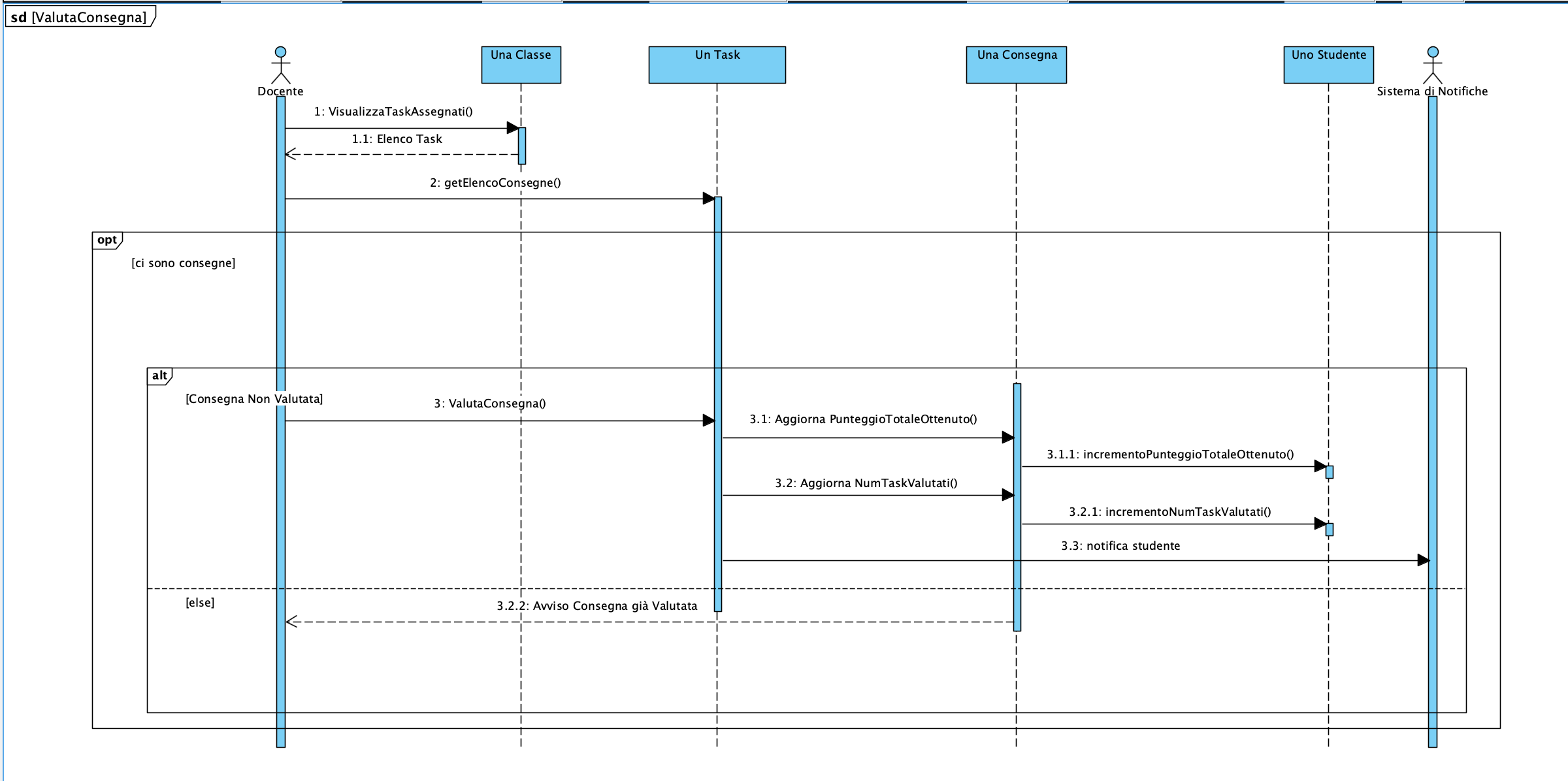
### ConsegnaSoluzione

La creazione del suddetto sequence diagram, sviluppato a partire dalla descrizione dello scenario del caso d’uso *ConsegnaSoluzione*, ha fatto sorgere la necessità di definire un attributo, specifico per la classe **Studente**, **numTaskValutati**, privato, per consentire un corretto calcolo della media e della classifica sul numero dei task completati.



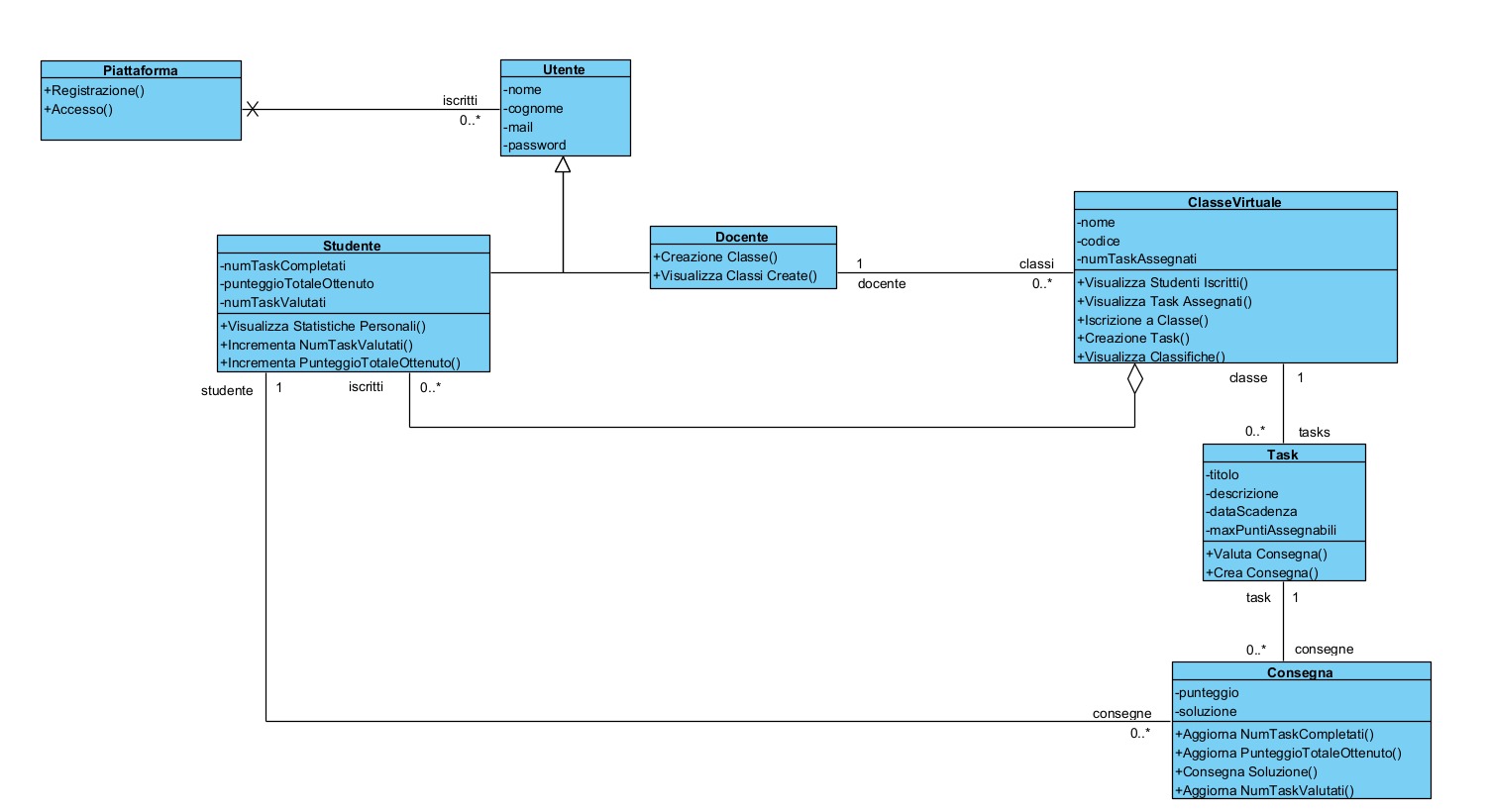
### ValutaConsegna

Per le stesse ragioni, è stato necessario inserire all’interno della classe **Consegna** il metodo **Aggiorna NumTaskValutati()** e i metodi **Incrementa NumTaskValutati() e Incrementa PunteggioTotaleOttenuto()** all’interno della classe **Studente**.



## Diagramma delle classi raffinato

Le aggiunte e le modifiche fatte nel corso della costruzione dei Sequence Diagrams hanno determinato lo sviluppo di un **Diagramma delle Classi** raffinato che riporta maggiori dettagli sugli attributi e le principali operazioni delle classi:



# Piano di test funzionale

Si intende progettare i casi di test funzionale con la tecnica del ***Category Partition Testing***.

## AssegnaTaskAClasse

|  |  |  |  |
| --- | --- | --- | --- |
| **AssegnaTaskAClasse** | | | |
| **TITOLO** | **DESCRIZIONE** | **DATA\_SCADENZA** | **MAX\_PUNTI\_ASSEGNABILI** |
| * Stringa di caratteri di lunghezza <= 25 * Stringa di caratteri di lunghezza > 25 **[ERROR]** | * Stringa di caratteri di lunghezza <= 200 * Stringa di caratteri di lunghezza > 200 **[ERROR]** | * Stringa di caratteri nel formato data yyyy-mm-dd * Stringa di caratteri non nel formato data yyyy-mm-dd   **[ERROR]**   * Stringa di caratteri nel formato data yyyy-mm-dd di una data non esistente   **[ERROR]**   * Stringa di caratteri nel formato data yyyy-mm-dd di una data precedente alla data di inserimento **[ERROR]** | * 1 <= Intero <= 100 * Intero < 1 **[ERROR]** * Intero > 100 **[ERROR]** * Non un numero intero **[ERROR]** |

Il numero di test da effettuarsi senza particolari vincoli è: .

Con i vincoli [ERROR], invece, il numero di test da eseguire per testare singolarmente i vincoli è 8 (1 per Titolo, 1 per Descrizione, 3 per Data\_Scadenza, 3 per Max\_Punti\_Assegnabili).

Il numero di test risultante è: (1\*1\*1\*1) + 8 = 9.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TEST SUITE** | | | | | | |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output attesi** | **Post-condizioni attese** |
| 1 | Tutti input validi | Titolo,  Descrizione,  Data\_Scadenza,  Max\_Punti\_Assegnabili  Validi |  | { Titolo: “Homework1”,  Descrizione: “Svolgere il compito dato in classe”, Data\_Scadenza: “2026-08-16”, Max\_Punti\_Assegnabili: “10” } | Task creato con successo. | Il Task creato è ora visibile nell’elenco dei task della classe |
| 2 | Titolo stringa > 25 caratteri | Titolo stringa di caratteri di lunghezza > 25 [ERROR],  Descrizione,  Data\_Scadenza,  Max\_Punti\_Assegnabili  Validi |  | { Titolo: “Homework1aaaaaaaaaaaaaaaaa”,  Descrizione: “Svolgere il compito dato in classe”, Data\_Scadenza: “2026-08-16”, Max\_Punti\_Assegnabili: “10” } | Il titolo non può essere più lungo di 25 caratteri. |  |
| 3 | Descrizione stringa > 200 caratteri | Descrizione stringa di caratteri di lunghezza > 200 [ERROR]  Titolo,  Data\_Scadenza,  Max\_Punti\_Assegnabili  Validi |  | { Titolo: “Homework1”,  Descrizione: “Stringa di 220 caratteri”, Data\_Scadenza: “2026-08-16”, Max\_Punti\_Assegnabili: “10” } | La descrizione non può essere più lunga di 200 caratteri |  |
| 4 | Data\_Scadenza stringa di caratteri non nel formato data yyyy-mm-dd | Data\_Scadenza stringa di caratteri non nel formato data yyyy-mm-dd [ERROR]  Titolo,  Descrizione,  Max\_Punti\_Assegnabili  Validi |  | { Titolo: “Homework1”,  Descrizione: “Svolgere il compito dato in classe”, Data\_Scadenza: “16u08@26”, Max\_Punti\_Assegnabili: “10” } | La data non è formattata correttamente |  |
| 5 | Data\_Scadenza stringa di caratteri nel formato data yyyy-mm-dd di una data non esistente | Data\_Scadenza stringa di caratteri nel formato data yyyy-mm-dd di una data non esistente [ERROR]  Titolo,  Descrizione,  Max\_Punti\_Assegnabili  Validi |  | { Titolo: “Homework1”,  Descrizione: “Svolgere il compito dato in classe”, Data\_Scadenza: “2026-02-31”, Max\_Punti\_Assegnabili: “10” } | Non è possibile inserire una data non esistente o passata. |  |
| 6 | Data\_Scadenza Stringa di caratteri nel formato data yyyy-mm-dd di una data precedente alla data di inserimento | Data precedente alla data di inserimento [ERROR]  Titolo,  Descrizione,  Max\_Punti\_Assegnabili  Validi | Data odierna = 16/08/2025 | { Titolo: “Homework1”,  Descrizione: “Svolgere il compito dato in classe”, Data\_Scadenza: “2024-07-19”, Max\_Punti\_Assegnabili: “10” } | Non è possibile inserire una data non esistente o passata. |  |
| 7 | Max\_Punti\_Assegnabili < 1 | Max\_Punti\_Assegnabili  Validi intero < 1 [ERROR]  Titolo,  Descrizione,  Data\_Scadenza |  | { Titolo: “Homework1”,  Descrizione: “Svolgere il compito dato in classe”, Data\_Scadenza: “2026-08-16”, Max\_Punti\_Assegnabili: “-1” } | Non è possibile assegnare un punteggio non positivo o superiore a 100. |  |
| 8 | Max\_Punti\_Assegnabili > 100 | Max\_Punti\_Assegnabili  Validi intero > 100 [ERROR]  Titolo,  Descrizione,  Data\_Scadenza |  | { Titolo: “Homework1”,  Descrizione: “Svolgere il compito dato in classe”, Data\_Scadenza: “2026-08-16”, Max\_Punti\_Assegnabili: “120” } | Non è possibile assegnare un punteggio non positivo o superiore a 100. |  |
| 9 | Max\_Punti\_Assegnabili diverso da numero intero | Max\_Punti\_Assegnabili  Validi non è un numero intero [ERROR]  Titolo,  Descrizione,  Data\_Scadenza |  | { Titolo: “Homework1”,  Descrizione: “Svolgere il compito dato in classe”, Data\_Scadenza: “2026-08-16”, Max\_Punti\_Assegnabili: “7.8” } | È stato inserito un tipo non conforme per il punteggio del Task. Inserire un valore numerico intero. |  |
| 10 | Max\_Punti\_Assegnabili diverso da numero intero | Max\_Punti\_Assegnabili  Validi non è un numero intero [ERROR]  Titolo,  Descrizione,  Data\_Scadenza |  | { Titolo: “Homework1”,  Descrizione: “Svolgere il compito dato in classe”, Data\_Scadenza: “2026-08-16”, Max\_Punti\_Assegnabili: “a” } | È stato inserito un tipo non conforme per il punteggio del Task. Inserire un valore numerico intero. |  |

## ConsegnaSoluzione

|  |
| --- |
| **ConsegnaSoluzione** |
| **SOLUZIONE** |
| * Stringa di caratteri di lunghezza <= 300 e >0 * Stringa di caratteri di lunghezza > 300**[ERROR]** * Stringa di caratteri di lunghezza = 0**[ERROR]** |

Il numero di test da effettuarsi senza particolari vincoli è:

Con i vincoli [ERROR], invece, il numero di test da eseguire per testare singolarmente i vincoli è 2 per Soluzione.

Il numero di test risultante è: 1 + 2 = 3.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST SUITE** | | | | | | | | | | | |
| **Test Case ID** | | | | | | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output attesi** | **Post-condizioni attese** |
| 1 | | | | | | Tutti input validi | Soluzione Valida |  | {Soluzione: “Stringa di 200 caratteri”} | Soluzione Consegnata con successo | Consegna aggiunta alla lista delle consegne di un determinato Task |
| 2 | Soluzione stringa > 300 caratteri | Soluzione stringa > 300 caratteri [ERROR] |  | {Soluzione: “Stringa di 301 caratteri”} | La soluzione è troppo lunga |  | | | | | |
| 3 | | | | | | Soluzione stringa vuota (lunghezza = 0) | Soluzione stringa vuota  [ERROR] |  | {Soluzione: “”} | La soluzione è vuota |  |

## ValutaConsegna

|  |
| --- |
| **ValutaConsegna** |
| **PUNTEGGIO** |
| * Punteggio con valori ≥0 e ≤Massimo Punteggio del relativo task * Punteggio con valori < 0 **[ERROR]** * Punteggio con valori > Massimo Punteggio del relativo task **[ERROR]** * Punteggio che contiene simboli che non sono valori numerici interi **[ERROR]** |

Il numero di test da effettuarsi senza particolari vincoli è: .

Con i vincoli [ERROR], invece, il numero di test da eseguire per testare singolarmente i vincoli è 3 (3 per punteggio).

Il numero di test risultante è: (1\*1) + 3 = 4.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TEST SUITE** | | | | | | |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | **Pre-condizioni** | **Input** | **Output attesi** | **Post-condizioni attese** |
| 1 | Tutti input validi | Punteggio valido | Punteggio Massimo del Task = 30 | {Punteggio: “27”} | Valutazione effettuata con successo | Lo studente viene notificato della valutazione |
| 2 | Punteggio <0 | Punteggio con valori<0 [ERROR] |  | {Punteggio: “-5”} | La valutazione è più alta del punteggio massimo assegnabile per questa consegna o è negativa |  |
| 3 | Punteggio > Massimo Punteggio del relativo task | Punteggio con valori > Massimo Punteggio del relativo task [ERROR] | Punteggio Massimo del Task = 10 | {Punteggio: “12”} | La valutazione è più alta del punteggio massimo assegnabile per questa consegna o è negativa |  |
| 4 | Punteggio contenente simboli non numerici interi | Punteggio che contiene simboli che non sono valori interi [ERROR] |  | {Punteggio: “a”} | È stato inserito un tipo non conforme per la valutazione. |  |

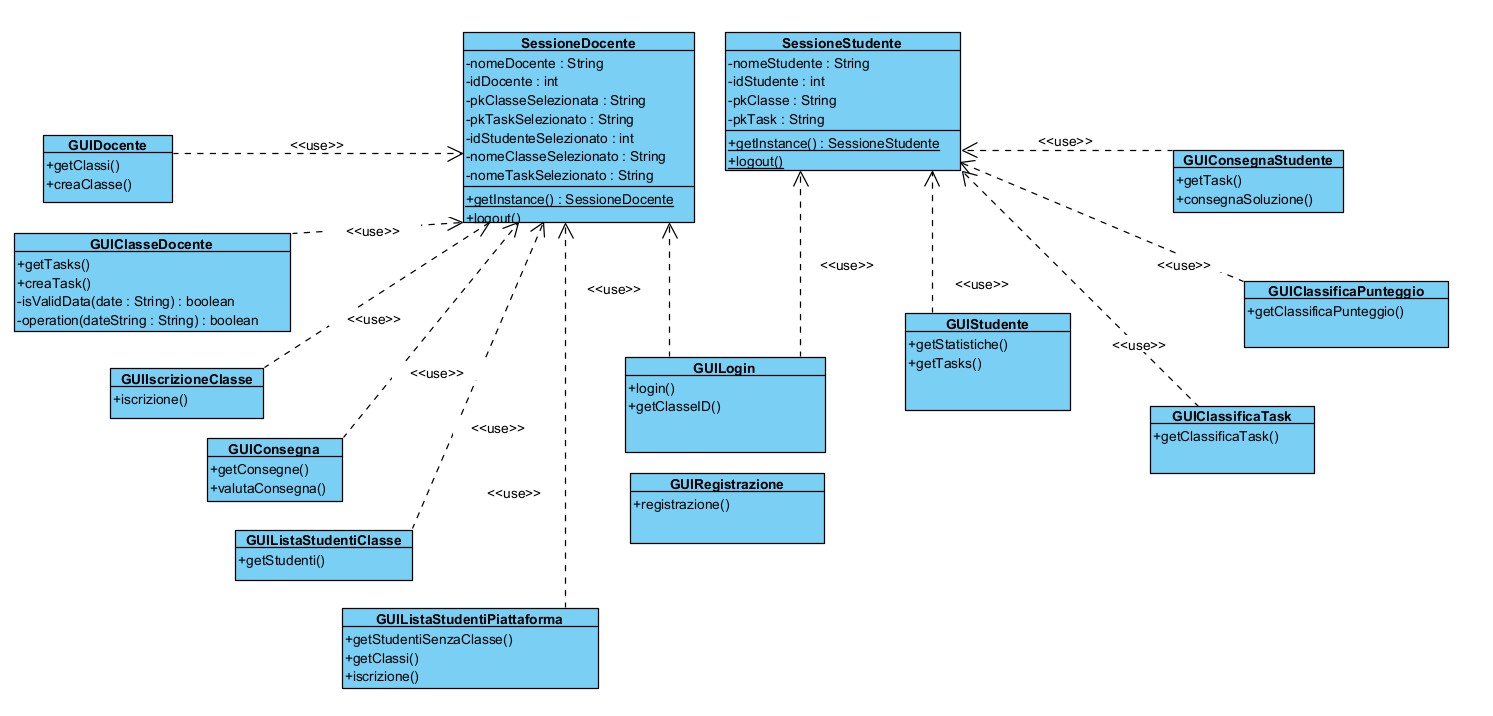
# Progettazione

## Diagramma delle classi

### Pattern BCED

#### Package Boundary

Il package Boundary contiene tutti gli oggetti responsabili dell’interfaccia utente e della logica di presentazione; a questo livello tutte le classi corrispondono a delle interfacce e i relativi attributi non sono altro che gli elementi che le compongono, visualizzati a video.



#### Package Controller

Questo package contiene gli oggetti che percepiscono gli eventi generati dalle interazioni con l’interfaccia utente e ne demandano la gestione all’unico componente del sistema software responsabile della gestione della Business Logic, il package Entity.

Immagine che contiene testo, schermata, Carattere, numero

Il contenuto generato dall'IA potrebbe non essere corretto.

#### Package Entity

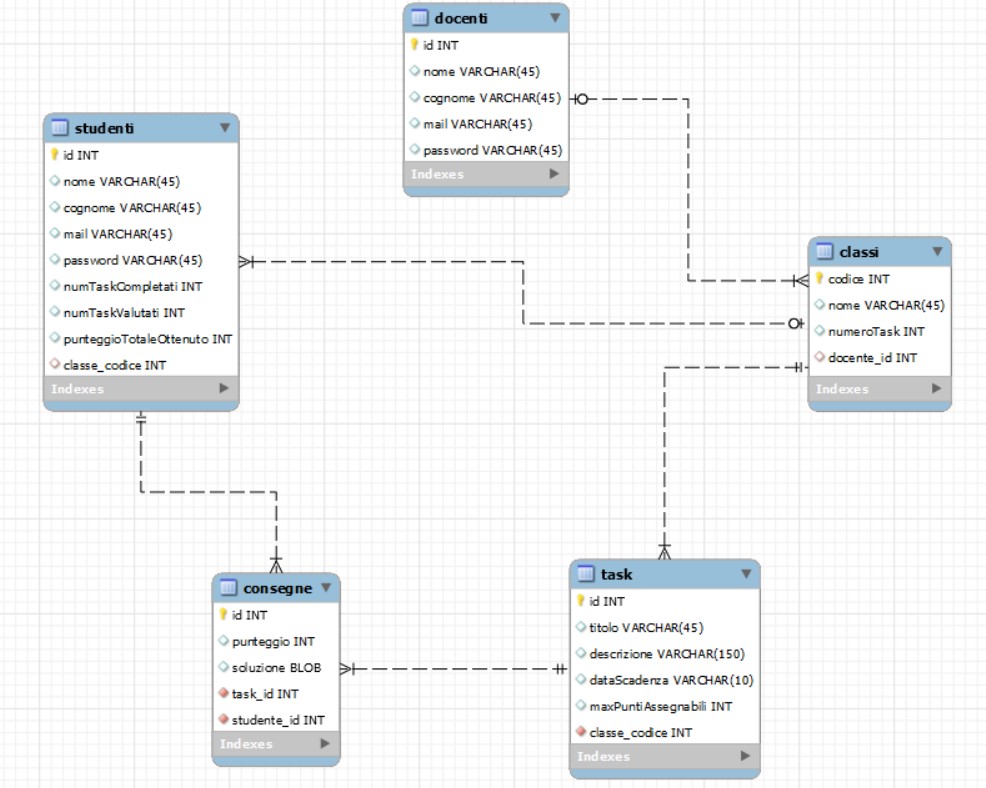
Il Package Entity contiene tutti gli oggetti che rappresentano la semantica delle entità del dominio applicativo e corrispondono alle strutture dati presenti all’interno del database di persistenza.

Immagine che contiene testo, schermata, Carattere, diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.

#### Modello E/R

Per poter introdurre l’ultimo package, risulta opportuno elencare qui di seguito il modello E/R (Entity-Relationship), che mostra le scelte progettuali fatte in termini di quali dati memorizzare nel Database, così come le relazioni implementate tra essi.



Questa rappresentazione grafica risulta utile per comprendere le scelte fatte in termini di **Traduzione dalle tabelle del Database Relazionale alle classi del Package Database**.

In particolare, dato che nel Modello E/R non compaiono relazioni n:m, ma esclusivamente relazioni di tipo 1:n, la traduzione effettuata ha consentito di individuare una classe univoca del package Database per ognuna delle tabelle del Database. Infine, per semplificare l’accesso al Database e migliorarne le prestazioni, nella traduzione della relazione di Generalizzazione-Specializzazione tra le Classi *Utente-Studente-Docente,* anziché adottare 3 tabelle distinte si è preferito tradurre in tabelle solo le classi figlie, inserendo al loro interno anche gli attributi di *Utente.*

#### Package Database

Da questo punto di vista, il Package Database contiene tutte le classi responsabili dell’estrazione dei dati dal DB, esponendo una vera e propria interfaccia che di fatto rende indipendenti le classi della Business Logic (Entity) dalla tecnologia di persistenza utilizzata.

In particolare, tra le strategie di risoluzione del problema dell’**impedance mismatch**, che nasce dalla mancata corrispondenza tra il modello Object Oriented e quello relazionale, si è deciso di adottare quella delle classi **DAO** (Data Access Objects), che consiste nell’utilizzo di appositi oggetti per l’accesso ai dati.

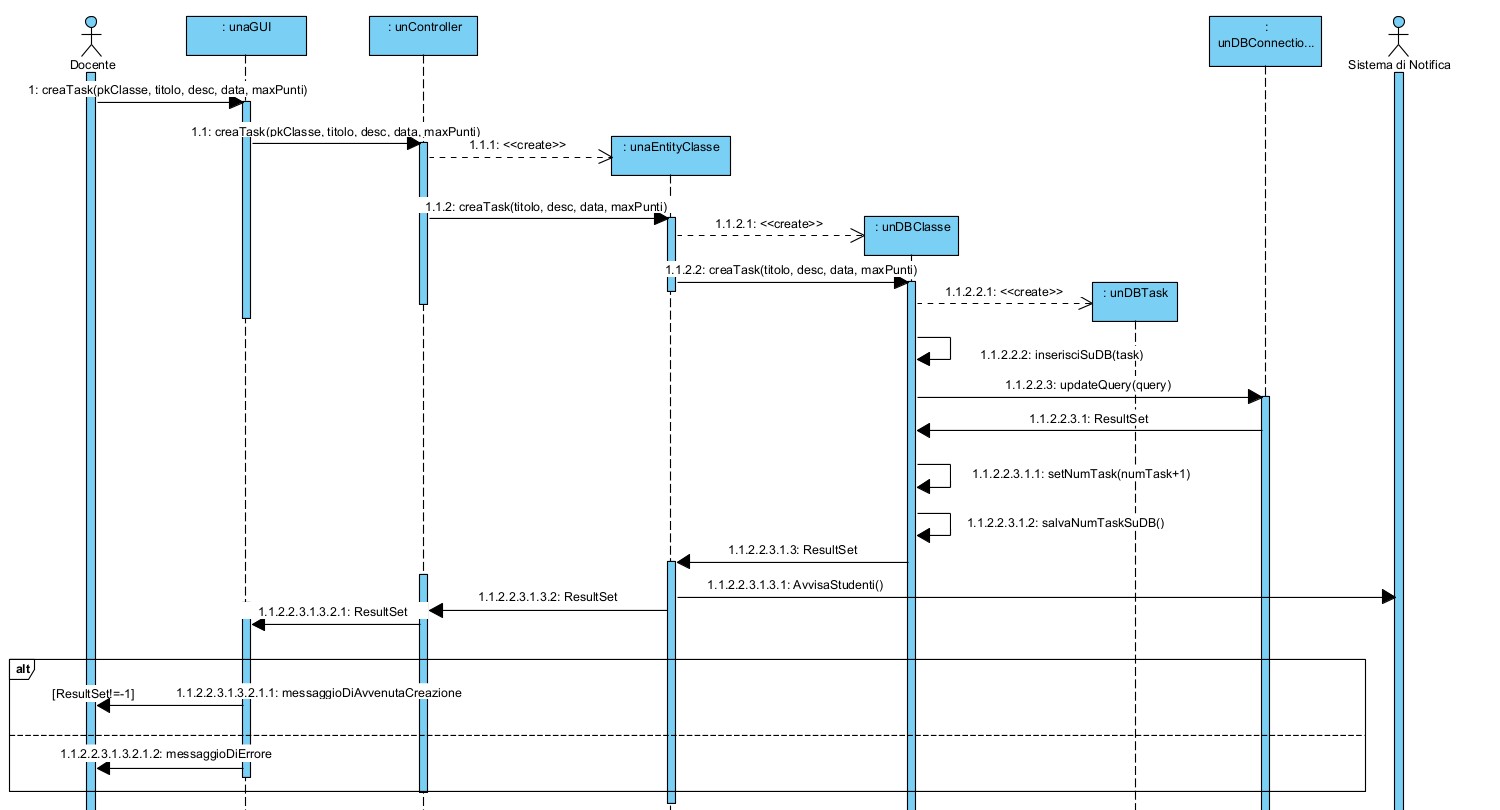
Ognuna di queste classi conterrà i metodi CRUD per l’interrogazione e la manipolazione della corrispondente classe di dominio (*query*), implementati in funzione di un’ulteriore classe, ***DBConnectionManager***, che costituisce di fatto l’unico punto di accesso vero e proprio al DB, sfruttando i metodi che questa mette a disposizione.

Immagine che contiene testo, schermata, diagramma, Rettangolo

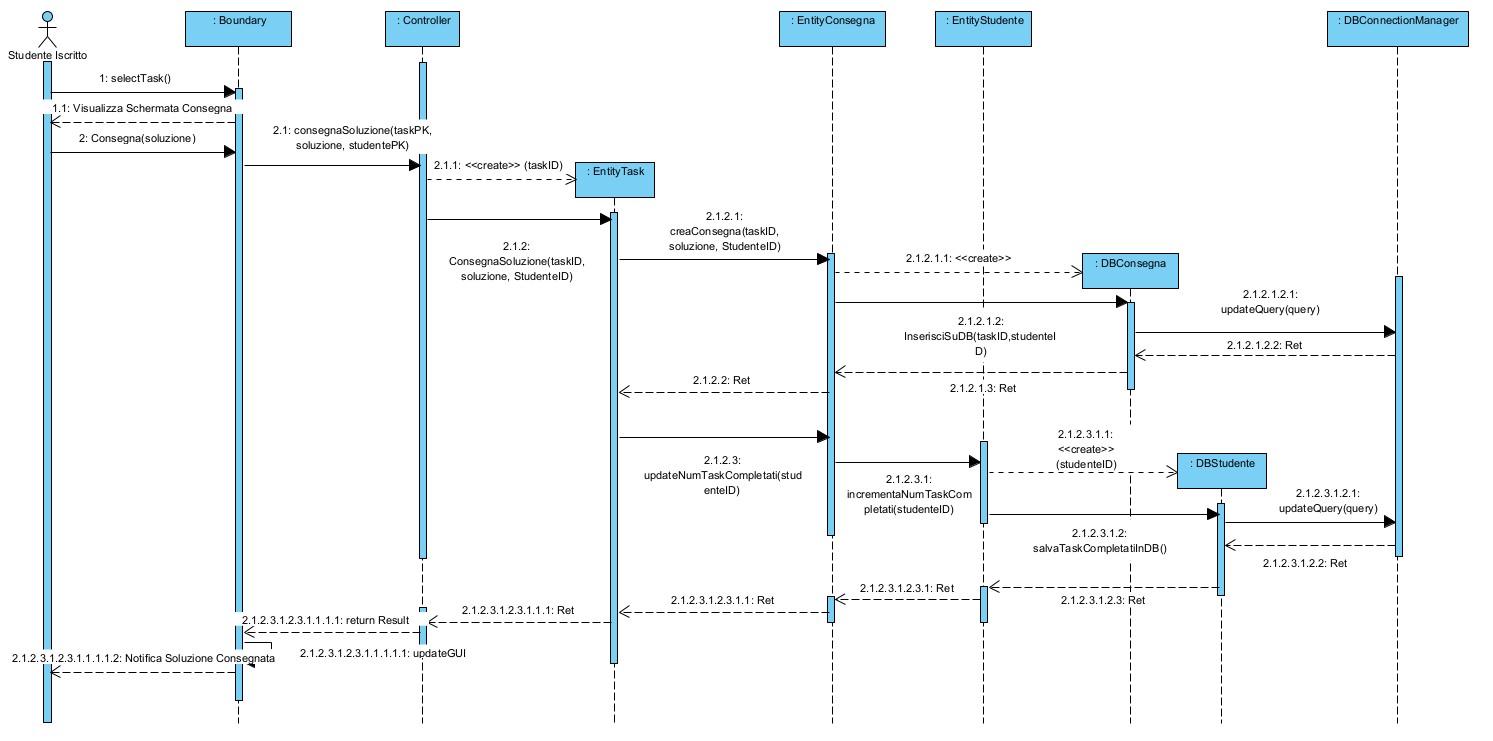
Il contenuto generato dall'IA potrebbe non essere corretto.

## Diagrammi di sequenza

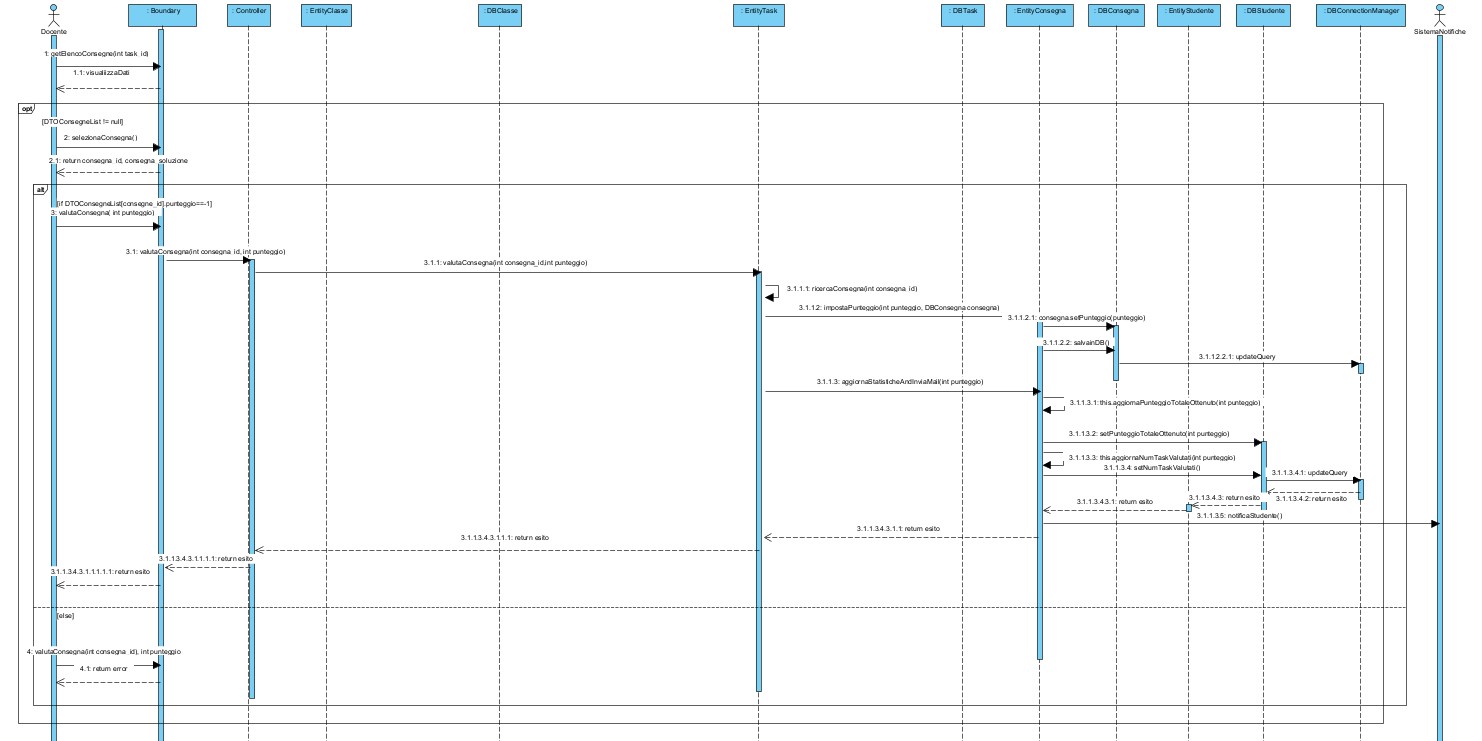
### AssegnaTaskAClasse



### ConsegnaSoluzione



### ValutaConsegna



I sequence progettati sono stati fondamentali per la corretta implementazione dell’applicazione software ed ha fatto nascere la necessità di definire ulteriori classi, metodi e funzioni, che hanno arricchito passo dopo passo il **Diagramma delle Classi di Progettazione**, fino ad ottenere la seguente versione finale:

Immagine che contiene testo, schermata, Carattere, diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.

# Implementazione

## Package Database

* DBConnectionManager
* DBUtente
* DBDocente
* DBStudente
* DBTask
* DBClasse
* DBConsegna

## Package Entity

* Piattaforma
* Utente
* EntityStudente
* EntityDocente
* EntityClasse
* EntityTask
* EntityConsegna

## Package Controller

* Controller

## Package Boundary

* SessioneDocente
* SessioneStudente
* GUIDocente
* GUIClasseDocente
* GUIIscrizioneClasse
* GUIConsegna
* GUIListaStudentiClasse
* GUIListaStudentiPiattaforma
* GUILogin
* GUIRegistrazione
* GUIStudente
* GUIClassificaTask
* GUIClassificaPunteggio
* GUIConsegnaStudente

## Package DTO

L’introduzione di tale package, estraneo al pattern BCED, nasce dall’esigenza di mostrare sulla GUI collezioni di elementi.

Da questo punto di vista, il problema principale è proprio quello che, qualora una determinata chiamata a funzione restituisse alla GUI un elenco di entity, questa, per poterlo visualizzare correttamente a video, dovrebbe conoscere di fatto la struttura interna di tale classe Entity, ma ciò porterebbe con sé un accoppiamento troppo elevato e quindi una chiara violazione dei vincoli del pattern a livelli adottato.

Si introduce allora il concetto di **Data Transfer Object** (**DTO**), un oggetto in grado di trasportare dati tra processi (nel caso in oggetto tra livelli). Le classi DTO hanno tipicamente una struttura che rispecchia quella dell’entity di cui vanno a supporto, in particolare gli attributi coincidono con quelli dell’entity che si intendono visualizzare a schermo.

* DTOUtente
* DTOStudente
* DTODocente
* DTOClasse
* DTOTask
* DTOConsegna

## Package Exceptions

* ClassCreationException
* ClassEnrollmentException
* DBException
* DescriptionLengthException
* FormattedDateException
* InvalidDateException
* MailAlreadyUsedException
* MailNotValidException
* RoleNotValidException
* ScoreEvaluationException
* SistemaNotificheException
* StudentException
* SubmissionAlreadyEvaluatedException
* SubmissionEvaluationException
* SubmissionException
* SubmissionExistingException
* TaskCreationException
* TitleLengthException
* UnassignableTaskScoreException

## Dipendenze: Jakarta

Per poter usufruire del servizio mail esterno si è resa necessaria l’introduzione di una serie di 3 file .jar di dipendenze per la libreria Jakarta, che consente la comunicazione col server esterno SMTP. In particolare, le dipendenze sono le seguenti:

* jakarta.activation-2.0.1
* jakarta.activation-api-2.0.1
* jakarta.mail-2.0.1

## Diagramma di Deployment

I diagrammi di deployment sono utilizzati per mostrare l’architettura fisica del sistema software realizzato; sono particolarmente utili per valutare, durante lo sviluppo, come un’applicazione si distribuisce tra le varie macchine.

Immagine che contiene testo, schermata, diagramma, Rettangolo

Il contenuto generato dall'IA potrebbe non essere corretto.

# Testing

## Test funzionale

Segue una descrizione in forma tabellare dei risultati dell’esecuzione dei test funzionali precedentemente pianificati.

|  |  |  |
| --- | --- | --- |
| **AssegnaTaskAClasse** | | |
| **TEST SUITE** | | | | | | | | | | |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | | **Pre-condizioni** | **Input** | **Output attesi** | **Post-condizioni attese** | **Output ottenuti** | **Post-condizioni ottenute** | **Esito (FAIL, PASS)** |
| 1 | Tutti input validi | Titolo,  Descrizione,  Data Scadenza,  Max\_Punti\_Assegnabili  Validi | |  | {Titolo:”Homework1”,  Descrizione: “Svolgere il compito dato in classe”,  Data\_scadenza:”2026-08-16”, Max\_Punti\_Assegnabili:”10”} | Task creato con successo. | Task creato e ora visibile nell’elenco della classe | Task creato con successo. | Task creato e ora visibile nell’elenco della classe | **PASS** |
| 2 | Titolo stringa>25 caratteri | Titolo stringa di caratteri di lunghezza>25 [ERROR], Descrizione, Data\_Scadenza,  Max\_Punti\_Assegnabili  Validi | |  | {Titolo:”Homework1aaaaaaaaaaaaaaaaaaaaaaaaaaaaa”,  Descrizione: “Svolgere il compito dato in classe”,  Data\_scadenza:”2026-08-16”, Max\_Punti\_Assegnabili:”10”} | Il titolo non può essere più lungo di 25 caratteri. |  | Il titolo non può essere più lungo di 25 caratteri. |  | **PASS** |
| 3 | Descrizione stringa>200 caratteri | Descrizione stringa di caratteri di lunghezza>200 [ERROR],  Titolo Descrizione, Data\_Scadenza,  Max\_Punti\_Assegnabili  Validi | |  | {Titolo:”Homework”,  Descrizione: “Stringa di 220 caratteri”,  Data\_scadenza:”2026-08-16”, Max\_Punti\_Assegnabili:”10”} | La descrizione non può essere più lunga di 200 caratteri |  | La descrizione non può essere più lunga di 200 caratteri |  | **PASS** |
| 4 | Data\_Scadenza stringa di caratteri non nel formato data yyyy-mm-dd | Data\_Scadenza stringa di caratteri non nel formato data yyyy-mm-dd [ERROR]  Titolo,  Descrizione,  Max\_Punti\_Assegnabili  Validi | |  | {Titolo:”Homework”,  Descrizione: ““Svolgere il compito dato in classe”,  Data\_scadenza:”16u08@26”, Max\_Punti\_Assegnabili:”10”} | La data non è formattata correttamente |  | La data non è formattata correttamente |  | **PASS** |
| 5 | Data\_scadenza stringa di caratterinel formato data yyyy-mm-dd di una data non esistente | Data\_Scadenza stringa di caratteri nel formato data yyyy-mm-dd di una data non esistente [ERROR]  Titolo,  Descrizione,  Max\_Punti\_Assegnabili  Validi | |  | {Titolo:”Homework”,  Descrizione: ““Svolgere il compito dato in classe”,  Data\_scadenza:”2026-02-31”, Max\_Punti\_Assegnabili:”10”} | Non è possibile inserire una data non esistente o passata. |  | Non è possibile inserire una data non esistente o passata. |  | **PASS** |
| 6 | Data\_scadenza stringa di caratterinel formato data yyyy-mm-dd di una data precedente alla data di inserimento | Data precedente alla data di inserimento [ERROR]  Titolo,  Descrizione,  Max\_Punti\_Assegnabili  Validi | | Data odierna=2025-06-05 | {Titolo:”Homework”,  Descrizione: ““Svolgere il compito dato in classe”,  Data\_scadenza:”2024-07-19”, Max\_Punti\_Assegnabili:”10”} | Non è possibile inserire una data non esistente o passata. |  | Non è possibile inserire una data non esistente o passata. |  | **PASS** |
| 7 | Max\_Punti\_Assegnabili<1 | Max\_Punti\_Assegnabili  Validi intero < 1 [ERROR]  Titolo,  Descrizione,  Data\_Scadenza | |  | { Titolo: “Homework1”,  Descrizione: “Svolgere il compito dato in classe”, Data\_Scadenza: “2026-08-16”, Max\_Punti\_Assegnabili: “-1” } | Non è possibile assegnare un punteggio non positivo o superiore a 100. |  | Non è possibile assegnare un punteggio non positivo o superiore a 100. |  | **PASS** |
| 8 | Max\_Punti\_Assegnabili>100 | Max\_Punti\_Assegnabili>100[ERROR]  Titolo,  Descrizione,  Data\_Scadenza  Validi | |  | { Titolo: “Homework1”,  Descrizione: “Svolgere il compito dato in classe”, Data\_Scadenza: “2026-08-16”, Max\_Punti\_Assegnabili: “120” } | Non è possibile assegnare un punteggio non positivo o superiore a 100. |  | Non è possibile assegnare un punteggio non positivo o superiore a 100. |  | **PASS** |
| 9 | Max\_Punti\_Assegnabili diverso da numero intero | Max\_Punti\_Assegnabili  Validi non è un numero intero [ERROR]  Titolo,  Descrizione,  Data\_Scadenza | |  | { Titolo: “Homework1”,  Descrizione: “Svolgere il compito dato in classe”, Data\_Scadenza: “2026-08-16”, Max\_Punti\_Assegnabili: “7.8” } | È stato inserito un tipo non conforme per il punteggio del Task. Inserire un valore numerico intero. |  | È stato inserito un tipo non conforme per il punteggio del Task. Inserire un valore numerico intero. |  | **PASS** |
| 10 | Max\_Punti\_Assegnabili diverso da numero intero | Max\_Punti\_Assegnabili  Validi non è un numero intero [ERROR]  Titolo,  Descrizione,  Data\_Scadenza | |  | { Titolo: “Homework1”,  Descrizione: “Svolgere il compito dato in classe”, Data\_Scadenza: “2026-08-16”, Max\_Punti\_Assegnabili: “a” } | È stato inserito un tipo non conforme per il punteggio del Task. Inserire un valore numerico intero. |  | È stato inserito un tipo non conforme per il punteggio del Task. Inserire un valore numerico intero. |  | **PASS** |

|  |  |  |
| --- | --- | --- |
| **ConsegnaSoluzione** | | |
| **TEST SUITE** | | | | | | | | | | |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | | **Pre-condizioni** | **Input** | **Output attesi** | **Post-condizioni attese** | **Output ottenuti** | **Post-condizioni ottenute** | **Esito (FAIL, PASS)** |
| 1 | Tutti input validi | Soluzione Valida | |  | {Soluzione: “Stringa di  200 caratteri”} | Soluzione Consegnata  con successo | Consegna  aggiunta alla  lista delle  consegna di  un determinato Task. | Soluzione Consegnata  con successo | Consegna  aggiunta alla  lista delle  consegna di  un determinato Task. | **PASS** |
| 2 | Soluzione stringa  > 300 caratteri | Soluzione stringa  > 300 caratteri [ERROR] | |  | {Soluzione: “Stringa di  301 caratteri”} | La soluzione è troppo lunga |  | La soluzione è troppo lunga |  | **PASS** |
| 3 | Soluzione stringa  > 300 caratteri | Soluzione stringa vuota (lunghezza = 0) | |  | {Soluzione: “”} | La soluzione è vuota |  | La soluzione è vuota |  | **PASS** |

|  |  |  |
| --- | --- | --- |
| **ValutaConsegna** | | |
| **TEST SUITE** | | | | | | | | | | |
| **Test Case ID** | **Descrizione** | **Classi di equivalenza coperte** | | **Pre-condizioni** | **Input** | **Output attesi** | **Post-condizioni attese** | **Output ottenuti** | **Post-condizioni ottenute** | **Esito (FAIL, PASS)** |
| 1 | Tutti input validi | Punteggio valido | | Punteggio Massimo del task=30 | {“Punteggio:27”} | Valutazione  effettuata con successo | Lo studente  viene  notificato  della  valutazione | Valutazione  effettuata con successo. | Lo studente  viene  notificato  della  valutazione | **PASS** |
| 2 | Punteggio <0 | Punteggio con  valori<0  [ERROR] | |  | {Punteggio: “-5”} | La valutazione è più alta del punteggio massimo assegnabile per questa consegna o è negativa |  | La valutazione è più alta del punteggio massimo assegnabile per questa consegna o è negativa |  | **PASS** |
| 3 | Punteggio >  Massimo  Punteggio del  relativo task | Punteggio con  valori > Massimo  Punteggio del  relativo task  [ERROR] | | Punteggio  Massimo  del Task =  10 | {Punteggio: “12”} | La valutazione è più alta del punteggio massimo assegnabile per questa consegna o è negativa |  | La valutazione è più alta del punteggio massimo assegnabile per questa consegna o è negativa |  | **PASS** |
| 4 | Punteggio contenente simboli non numerici interi | Punteggio contenente simboli non numerici interi [ERROR] | |  | {Punteggio: “a”} | È stato inserito un tipo non conforme per la valutazione. Inserire un valore numerico intero. |  | È stato inserito un tipo non conforme per la valutazione. Inserire un valore numerico intero. |  | **PASS** |