



UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II
CORSO DI LAUREA IN INGEGNERIA INFORMATICA
CORSO DI INGEGNERIA DEL SOFTWARE
PROF. A.R. FASOLINO - A.A. 2022 - 23

Progetto

Gestione Istituto Scolastico

Studenti: Federica Del Vecchio, N4600443,
feder.delvecchio@studenti.unina.it

Claudia Antonella Esposito, N43001590,
claudiaa.esposito@studenti.unina.it

Saida Benwadih, N46004908,
s.benwadih@studenti.unina.it

INDICE

1. Specifiche informali
2. Analisi e specifica dei requisiti.....
2.1 Analisi nomi-verbi.....
2.2 Revisione dei requisiti.....
2.3 Glossario dei termini.....
2.4 Classificazione dei requisiti.....
2.4.1 Requisiti funzionali
2.4.2 Requisiti sui dati.....
2.4.3 Vincoli / Altri requisiti.....
2.5 Modellazione dei casi d'uso
2.5.1 Attori e casi d'uso
2.5.2 Diagramma dei casi d'uso
2.5.3 Scenari.....
2.6 Diagramma delle classi.....
2.7 Diagrammi di sequenza.....
2.8 Verifica della completezza dei requisiti.....
3. Stima dei costi
4. Piano di test funzionale.....
5. Progettazione
5.1 Diagramma delle classi.....
5.2 Diagrammi di sequenza.....
6. Implementazione
7. Testing.....
7.1 Test strutturale
7.1.1 Complessità ciclomatica
7.2 Test funzionale

1. Specifiche informali

Si vuole realizzare sistema software per la gestione di un istituto scolastico.

La segreteria registra al sistema gli utenti, ovvero i docenti, gli alunni e i genitori, specificando il nome, cognome, data di nascita, codice fiscale, comune di residenza, e-mail, numero di cellulare, username, password.

L'istituto scolastico è organizzato in classi. Ogni classe è caratterizzata dalla sezione (identificata da una lettera) e da un anno espresso in numeri romani (da 1 a 5). Per ogni classe il sistema tiene traccia del registro elettronico (uno per ciascun anno scolastico), che può essere consultato dal preside e dai docenti.

Per i docenti si vuole tenere traccia delle classi ove insegnano e delle materie insegnate in tali classi in ciascun anno scolastico. Ogni alunno è caratterizzato anche da una matricola e dalla classe frequentata.

I docenti di ogni classe devono riportare nel registro le attività (descritte come testo libero) svolte in una certa classe in una certa data. Inoltre, essi possono aggiungere i voti di ogni alunno, riportando la data. Se il voto inserito dal docente è non sufficiente (< 6) viene inviata una notifica ai genitori dell'alunno che ha ottenuto la valutazione negativa.

Gli alunni di una classe possono consultare il registro per leggere le attività svolte in una determinata data, mentre i genitori possono visionare la media voti del proprio figlio nell'ultimo quadri mestre più il dettaglio dei singoli voti in tutte le materie.

Al termine di ogni quadri mestre, il sistema genera la pagella di ogni alunno riportando la media voti per ogni materia. I docenti possono apportare delle modifiche alle valutazioni relative ai propri insegnamenti arrotondando il voto per eccesso o per difetto.

Appena creata, la pagella si trova nello stato "non approvata": è compito del preside approvare la pagella. La pagella non può essere modificata dai docenti una volta approvata.

All'atto dell'approvazione, se tutte le pagelle associate agli alunni della classe sono state approvate, il sistema notifica tutti i genitori degli alunni, inviando loro una e-mail.

2. Analisi e specifica dei requisiti

2.1 Analisi nomi-verbi

Si vuole realizzare sistema software per la gestione di un istituto scolastico.

La segreteria registra al sistema gli utenti, ovvero i docenti, gli alunni e i genitori, specificando il nome, cognome, data di nascita, codice fiscale, comune di residenza, e-mail, numero di cellulare, username, password.

L'istituto scolastico è organizzato in classi. Ogni classe è caratterizzata dalla sezione (identificata da una lettera) e da un anno espresso in numeri romani (da 1 a 5). Per ogni classe il sistema tiene traccia del registro elettronico (uno per ciascun anno scolastico), che può essere consultato dal preside e dai docenti.

Per i docenti si vuole tenere traccia delle classi ove insegnano e delle materie insegnate in tali classi in ciascun anno scolastico. Ogni alunno è caratterizzato anche da una matricola e dalla classe frequentata.

I docenti di ogni classe devono riportare nel registro le attività (descritte come testo libero) svolte in una certa classe in una certa data. Inoltre, essi possono aggiungere i voti di ogni alunno, riportando la data. Se il voto inserito dal docente è non sufficiente (< 6) viene inviata una notifica ai genitori dell'alunno che ha ottenuto la valutazione negativa.

Gli alunni di una classe possono consultare il registro per leggere le attività svolte in una determinata data, mentre i genitori possono visionare la media voti del proprio figlio nell'ultimo quadrimestre più il dettaglio dei singoli voti in tutte le materie.

Al termine di ogni quadrimestre, il sistema genera la pagella di ogni alunno riportando la media voti per ogni materia. I docenti possono apportare delle modifiche alle valutazioni relative ai propri insegnamenti arrotondando il voto per eccesso o per difetto.

Appena creata, la pagella si trova nello stato "non approvata": è compito del preside approvare la pagella. La pagella non può essere modificata dai docenti una volta approvata.

All'atto dell'approvazione, se tutte le pagelle associate agli alunni della classe sono state approvate, il sistema notifica tutti i genitori degli alunni, inviando loro una e-mail.

LEGENDA:

Classe
Attributo
Funzionalità
Attore
Classe-Attore

2.2 Revisione dei requisiti

1. La segreteria registra al sistema gli utenti.
2. Gli utenti sono i docenti, gli alunni e i genitori.
3. Gli utenti sono caratterizzati da nome, cognome, data di nascita, codice fiscale, comune di residenza, e-mail, numero di cellulare, username, password.
4. L'istituto scolastico è organizzato in classi.
5. Ogni classe è caratterizzata da una sezione e da un anno.
6. La sezione della classe è identificata da una lettera.
7. L'anno della classe è espresso in numeri romani (da 1 a 5).
8. Il sistema tiene traccia del registro elettronico per ogni classe e per ciascun anno scolastico.

9. Il registro elettronico può essere consultato dal preside.
10. Il registro elettronico può essere consultato dai docenti.
11. Il sistema vuole tenere traccia per ogni docente e per ogni anno scolastico delle classi ove insegna e della materia insegnata in ogni classe.
12. Ogni alunno è caratterizzato da una matricola e dalla classe frequentata.
13. I docenti di ogni classe devono poter riportare nel registro le attività svolte in una certa classe in una certa data.
14. I docenti devono poter aggiungere i voti di ogni alunno, riportando la data.
15. Il voto inserito dal docente è considerato insufficiente se è minore di 6.
16. Il sistema invia una notifica ai genitori dell'alunno che ha ottenuto la valutazione negativa.
17. Gli alunni di una classe possono consultare il registro per leggere le attività svolte in una determinata data.
18. I genitori possono visionare la media voti del proprio figlio nell'ultimo quadrimestre più il dettaglio dei singoli voti in tutte le materie.
19. Il sistema, al termine di ogni quadrimestre, genera la pagella di ogni alunno riportando la media voti per ogni materia.
20. I docenti possono apportare delle modifiche alle valutazioni relative ai propri insegnamenti arrotondando il voto per eccesso o per difetto.
21. La pagella, appena creata, si trova nello stato "non approvata".
22. Il preside deve poter approvare la pagella.
23. La pagella non può essere modificata dai docenti una volta approvata.
24. Se tutte le pagelle associate agli alunni della classe sono state approvate, il sistema notifica tutti i genitori degli alunni, inviando loro una e-mail.

2.3 Glossario dei termini

Termine	Descrizione	Sinonimi
Segreteria	La segreteria rappresenta il personale scolastico responsabile della registrazione degli utenti.	
Utente	Gli utenti comprendono i docenti, gli alunni e i genitori.	
Classe	La classe è un insieme degli alunni che hanno in comune gli stessi docenti.	
Registro elettronico	Il registro permette ai docenti di inserire i dati relativi all'andamento scolastico degli alunni e alle attività svolte in classe.	Registro
Quadrimestre	Il quadrimestre è la suddivisione di un anno in periodi di quattro mesi l'uno.	
Anno scolastico	L'anno scolastico è composto da due quadrimestri.	
Voto	I docenti assegnano dei voti, che vanno da 0 a 10, agli alunni con lo scopo di valutarne la preparazione.	
Pagella	La pagella riepiloga la media dei voti e i singoli voti per ogni materia per ogni studente, al termine di ogni quadrimestre.	
Preside	Il preside è a capo di un istituto scolastico ed è incaricato di approvare le pagelle degli alunni.	Dirigente Scolastico

2.4 Classificazione dei requisiti

2.4.1 Requisiti funzionali

ID	Requisito	Origine (n. frase dei requisiti revisionati)
RF01	Il sistema deve offrire una funzionalità alla segreteria per poter registrare gli utenti.	1
RF02	Il sistema deve tener traccia del registro elettronico per ogni classe e per ciascun anno scolastico.	8
RF03	Il sistema deve offrire al preside una funzionalità per consultare il registro elettronico.	9
RF04	Il sistema deve offrire ai docenti una funzionalità per consultare il registro elettronico.	10
RF05	Il sistema deve tenere traccia per ogni docente e per ogni anno scolastico delle classi ove insegna e della materia insegnata in ogni classe.	11
RF06	Il sistema deve offrire una funzionalità per permettere al docente di poter registrare le attività svolte in una certa classe in una certa data.	13
RF07	Il sistema deve offrire una funzionalità per permettere al docente di poter aggiungere i voti di ogni alunno, specificando la data.	14
RF08	Il sistema deve inviare una notifica ai genitori dell'alunno che ha ottenuto la valutazione negativa.	16
RF09	Il sistema deve offrire una funzionalità agli alunni per permettergli di consultare il registro per leggere le attività svolte in una determinata data.	17
RF10	Il sistema deve offrire una funzionalità ai genitori per visionare la media voti del proprio figlio nell'ultimo quadrimestre più il dettaglio dei singoli voti in tutte le materie.	18
RF11	Il sistema, al termine di ogni quadrimestre, deve generare la pagella di ogni alunno riportando la media dei voti per ogni materia.	19
RF12	Il sistema deve offrire una funzionalità ai docenti per permettergli di apportare delle modifiche alle valutazioni relative ai propri insegnamenti arrotondando il voto per eccesso o per difetto.	20
RF13	Il sistema deve offrire una funzionalità al preside per permettergli di approvare una pagella.	22
RF14	Il sistema, se tutte le pagelle degli alunni della classe sono state approvate, deve notificare tutti i genitori, inviando loro una e-mail.	24

2.4.2 Requisiti sui dati

ID	Requisito	Origine (n. frase dei requisiti revisionati)
RD01	Gli utenti sono i docenti, gli alunni e i genitori.	2
RD02	Per ogni utente si vuole memorizzare nome, cognome, data di nascita, codice fiscale, comune di residenza, e-mail, numero di cellulare, username, password.	3
RD03	L'istituto scolastico è organizzato in classi.	4
RD04	Ogni classe è identificata dalla sezione e da un anno.	5
RD05	Ogni alunno è identificato da una matricola e dalla classe frequentata.	12

2.4.3 Vincoli / Altri requisiti

ID	Requisito	Origine (n. frase dei requisiti revisionati)
Vo1	La sezione della classe è una lettera.	6
Vo2	L'anno della classe è espresso in numeri romani, da 1 a 5.	7
Vo3	Il voto inserito dal docente è considerato insufficiente se è un numero minore di 6.	15
Vo4	La pagella, appena creata, si trova nello stato "non approvata".	21
Vo5	La pagella non può essere modificata dai docenti una volta approvata.	23

2.5 Modellazione dei casi d'uso

2.5.1 Attori e casi d'uso

Attori Primari:

- Segreteria
- Utente
- Docente
- Alunno
- Genitore
- Preside
- Tempo

Attori Secondari:

- Servizio E-mail

Casi d'uso

- UC1: Registrazione Utente
- UC2: Login
- UC3: Inserimento Attività
- UC4: Inserimento Voto
- UC5: Lettura Attività
- UC6: Lettura Carriera
- UC7: Generazione Pagella
- UC8: Modifica Pagella
- UC9: Approvazione Pagella

Casi d'uso di estensione:

- UC10: Invio E-mail Pagella
- UC11: Invio E-Mail Insufficienza

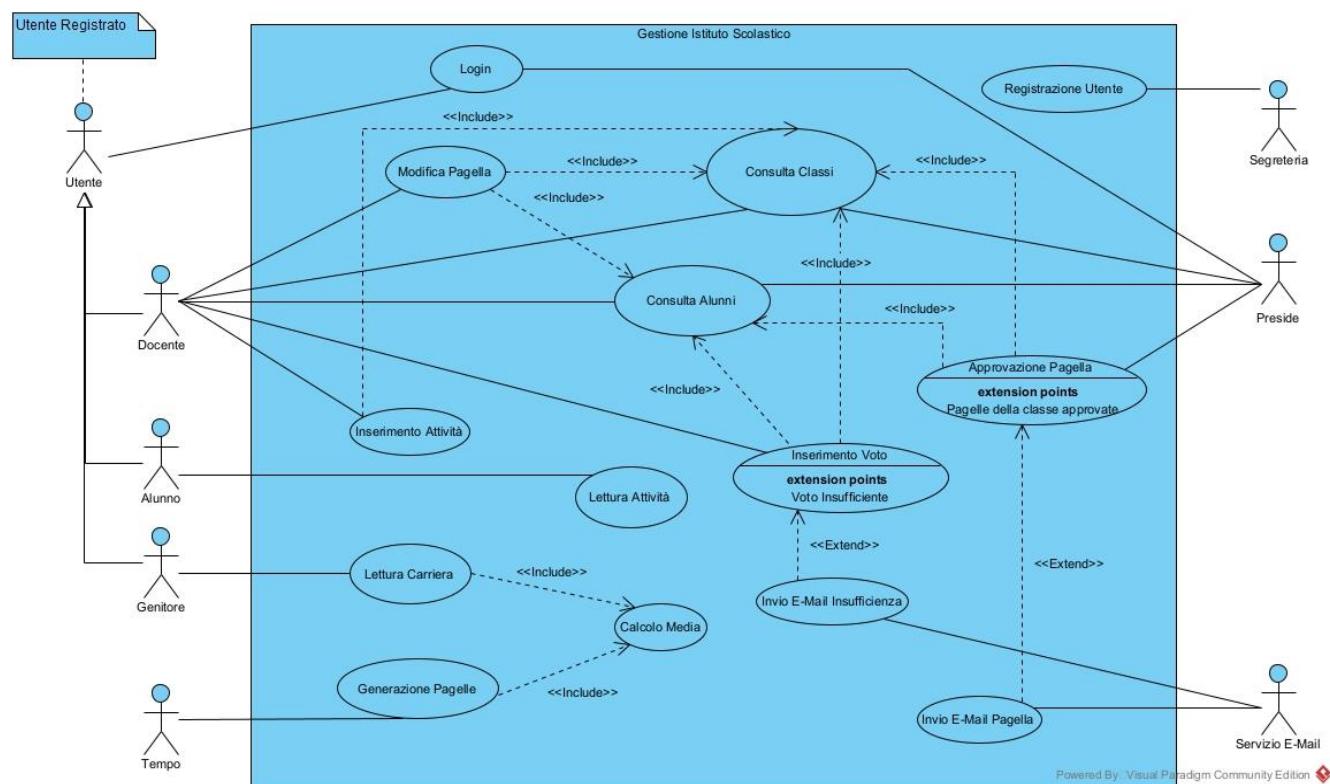
Casi d'uso di inclusione:

- UC12: Calcolo Media
- UC13: Consulta Classi
- UC14: Consulta Alunni

Caso d'uso	Attori Primari	Attori Secondari	Incl. / Ext.	Requisiti corrispondenti
UC1: Registrazione Utente	Segreteria	-	-	RF01
UC2: Login	Utente	-	-	-
UC3: Inserimento Attività	Docente	-	-	RF06
UC4: Inserimento Voto	Docente	-	-	RF07
UC5: Lettura Attività	Alunno	-	-	RF09
UC6: Lettura Carriera	Genitore	-	-	RF10
UC7: Generazione Pagella	Tempo	-	-	RF11
UC8: Modifica Pagella	Docente	-	-	RF12
UC9: Approvazione Pagella	Preside	-	-	RF13
UC10: Invio E-mail Pagella	-	Servizio E-mail	Estensione di Approvazione Pagella	RF14
UC11: Invia E-Mail Insufficienza	-	Servizio E-mail	Estensione di Inserimento Voti	RF08

UC12: Calcolo Media	-	Incluso in Lettura Carriera e Generazione Pagella	RF10, RF11
UC13: Consulta Classi	Preside, Docente	Incluso in Modifica Pagella, Inserisci Attività, Inserisci Voto, Approvazione Pagella	RF03, RF04
UC14: Consulta Alunni	Preside, Docente	Incluso in Modifica Pagella, Inserisci Voto, Approvazione Pagella	RF03, RF04

2.5.2 Diagramma dei casi d'uso



2.5.3 Scenari

Caso d'uso:	Registrazione Utente
Attore primario	Segreteria
Attore secondario	-
Descrizione	La segreteria registra un utente inserendo i dati nel sistema.
Pre-Condizioni	-
Sequenza di eventi principale	<ol style="list-style-type: none"> Il caso d'uso inizia quando la segreteria fa presente che vuole registrare un utente. Il sistema richiede l'inserimento dei dati necessari. La segreteria inserisce nome, cognome, data di nascita, codice fiscale, comune di residenza, e-mail, numero di cellulare, username e password dell'utente.

	4. IF un utente con l'username inserito non è già memorizzato, il sistema salva l'utente e le sue informazioni. 5. Altrimenti, il sistema restituisce un messaggio di ERRORE alla segreteria.
Post-Condizioni	Un nuovo utente risulta registrato.
Casi d'uso correlati	<i>Nessuno</i>
Sequenza di eventi alternativi	-

Caso d'uso: Login	
Attore primario	Utente, Preside
Attore secondario	-
Descrizione	Un utente o un preside accede al sistema dell'istituto scolastico.
Pre-Condizioni	L'utente deve essere stato registrato in precedenza mentre il preside deve già essere presente nel sistema.
Sequenza di eventi principale	1. Il caso d'uso inizia quando l'utente o il preside richiede l'autenticazione. 2. WHILE i dati non sono corretti 2.1. Il sistema richiede username e password. 2.2. L'utente inserisce i dati richiesti. 2.3. Il sistema controlla che i dati inseriti corrispondono a un utente o al preside registrato.
Post-Condizioni	I servizi offerti dal sistema sono ora accessibili all'utente o al preside.
Casi d'uso correlati	<i>nessuno</i>
Sequenza di eventi alternativi	-

Caso d'uso: Inserimento Attività	
Attore primario	Docente
Attore secondario	-
Descrizione	Un docente inserisce nel registro elettronico un'attività svolta per una certa materia in una certa classe in una certa data.
Pre-Condizioni	Il docente deve essere autenticato.
Sequenza di eventi principale	1. Il caso d'uso inizia quando il docente richiede l'inserimento di una nuova attività. 2. INCLUDE(Consulta Classi) 3. Il docente seleziona la materia e la classe. 4. Il sistema richiede l'inserimento della data e della descrizione. 5. Il docente inserisce la data e la descrizione dell'attività. 6. Il sistema memorizza l'attività nel registro elettronico della classe.
Post-Condizioni	Un'attività è stata aggiunta nel registro elettronico della classe.
Casi d'uso correlati	<i>Consulta Classi</i>
Sequenza di eventi alternativi	-

Caso d'uso: Inserimento Voto	
Attore primario	Docente
Attore secondario	-
Descrizione	Un docente inserisce nel registro elettronico uno nuovo voto per un alunno di una classe in una certa data e per una certa materia.

Pre-Condizioni	Il docente deve essere autenticato.
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il docente richiede l'inserimento di un voto. 2. INCLUDE(Consulta Classi) 3. INCLUDE(Consulta Alunni) 4. Il docente seleziona l'alunno. 5. Il sistema chiede il voto e la data. 6. WHILE il voto non è compreso tra 0 e 10 <ol style="list-style-type: none"> 6.1. Il docente inserisce il voto e la data. 7. IF il voto inserito è minore di 6 <ol style="list-style-type: none"> 7.1. PUNTO DI ESTENSIONE: Voto insufficiente 8. Il sistema salva la valutazione nel registro elettronico.
Post-Condizioni	Un nuovo voto è stato inserito nel registro elettronico della classe.
Casi d'uso correlati	<i>Invio E-Mail Insufficienza, Consulta Classi, Consulta Alunni</i>
Sequenza di eventi alternativi	-

Caso d'uso:	Lettura Attività
Attore primario	Alunno
Attore secondario	-
Descrizione	Un alunno visualizza le attività svolte in una certa data.
Pre-Condizioni	L'alunno deve aver fatto prima il login.
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'alunno richiede la visione delle attività svolte nella propria classe. 2. L'alunno inserisce la data di interesse. 3. Il sistema restituisce una lista che porta le descrizioni e le materie delle attività svolte nella giornata specificata.
Post-Condizioni	-
Casi d'uso correlati	<i>Nessuno</i>
Sequenza di eventi alternativi	-

Caso d'uso:	Lettura Carriera
Attore primario	Genitore
Attore secondario	-
Descrizione	Un genitore visualizza la carriera scolastica di un proprio figlio che è alunno dell'istituto. La carriera comprende sia la media che i singoli voti di ogni materia.
Pre-Condizioni	Il genitore deve essersi autenticato.
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il genitore richiede di poter accedere alla carriera. 2. IF il genitore ha più figli che fanno parte dell'istituto <ol style="list-style-type: none"> 2.1. Il sistema mostra una lista dei possibili alunni tra cui scegliere. 2.2. Il genitore seleziona l'alunno interessato. 3. INCLUDE(Calcola Media) 4. Il sistema stampa una lista contenente i voti e la loro media, compresi di materia.
Post-Condizioni	-
Casi d'uso correlati	<i>Calcola Media</i>
Sequenza di eventi alternativi	-

Caso d'uso:	Generazione Pagelle
Attore primario	Tempo
Attore secondario	-
Descrizione	Alla fine di ogni quadri mestre, il sistema genera automaticamente per ogni alunno una pagella contenente la media dei voti per ogni materia.
Pre-Condizioni	-
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia allo scadere del quadri mestre. 2. FOR ogni studente dell'istituto scolastico <ol style="list-style-type: none"> 2.1. INCLUDE(Calcola Media) 2.2. Il sistema genera la pagella dello studente inserendo la media dei voti per ogni materia.
Post-Condizioni	È stata creata una pagella per ogni studente dell'istituto.
Casi d'uso correlati	<i>Calcola Media</i>
Sequenza di eventi alternativi	-

Caso d'uso:	Modifica Pagella
Attore primario	Docente
Attore secondario	-
Descrizione	Un docente aggiorna la pagella di uno studente arrotondando la valutazione complessiva per eccesso o per difetto.
Pre-Condizioni	Le pagelle devono essere già state generate.
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il docente chiede di poter modificare una pagella specificando di voler eseguire un arrotondamento per eccesso o per difetto. 2. INCLUDE(Consulta Classi) 3. INCLUDE(Consulta Alunni) 4. Il docente seleziona l'alunno. 5. IF la pagella dell'alunno risulta già approvata <ol style="list-style-type: none"> 5.1. Il sistema restituisce un messaggio di ERRORE al docente. 6. Altrimenti, il sistema mostra la valutazione originale della media, apporta la modifica alla pagella e stampa il valore finale.
Post-Condizioni	La pagella è stata modificata.
Casi d'uso correlati	<i>Consulta Classi, Consulta Alunni</i>
Sequenza di eventi alternativi	-

Caso d'uso:	Approvazione Pagella
Attore primario	Preside
Attore secondario	-
Descrizione	Un preside approva la pagella di un alunno.
Pre-Condizioni	Le pagelle devono essere già state create.
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il preside decide di approvare una pagella. 2. INCLUDE (Consulta Classi) 3. INCLUDE(Consulta Alunni) 4. Il preside seleziona l'alunno. 5. IF la pagella non è già approvata <ol style="list-style-type: none"> 5.1. Il sistema approva la pagella dell'alunno specificato.

	<p>6. IF, le pagelle di tutti gli studenti facenti parte della classe dell'alunno selezionato sono approvate</p> <p>6.1 PUNTO DI ESTENSIONE: Pagelle della classe approvate</p>
Post-Condizioni	La pagella non può essere più aggiornata.
Casi d'uso correlati	<i>Invio E-Mail Pagella, Consulta Classi, Consulta Alunni</i>
Sequenza di eventi alternativi	-

Caso d'uso d'estensione: Invio E-Mail Insufficienza	
Attore primario	-
Attore secondario	Servizio E-Mail
Descrizione	Il servizio e-mail viene incaricato dal sistema di inviare una notifica al genitore dell'alunno che ha ricevuto una valutazione negativa.
Pre-Condizioni	Nel registro elettronico è stato recentemente inserito un voto insufficiente.
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il sistema rileva l'inserimento, nel registro di una classe, di un voto insufficiente per un alunno. 2. FOR per ogni genitore dello alunno <ol style="list-style-type: none"> 2.1. Il sistema comunica al servizio e-mail informazioni sul voto e l'alunno in questione e sul genitore da notificare, tra cui la sua e-mail.
Post-Condizioni	Il genitore riceve una e-mail sulla sua posta elettronica.
Casi d'uso correlati	<i>Inserimento Voto</i>
Sequenza di eventi alternativi	-

Caso d'uso d'estensione: Invio E-Mail Pagella	
Attore primario	-
Attore secondario	Servizio E-Mail
Descrizione	Il sistema incarica il servizio e-mail di inviare una notifica al genitore per informarlo del fatto che la pagella del figlio è pronta.
Pre-Condizioni	Tutte le pagelle degli studenti della classe di cui fa parte l'alunno devono esser state approvate.
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il sistema si accorge che le pagelle dell'alunno e dei suoi compagni di classe risultano approvate. 2. FOR ogni genitore degli alunni della classe <ol style="list-style-type: none"> 2.1. Il sistema comunica al servizio e-mail informazioni sulla pagella, l'alunno e il genitore, compresa la sua e-mail.
Post-Condizioni	Il genitore riceve l'e-mail sulla propria posta elettronica.
Casi d'uso correlati	<i>Approvazione Pagella</i>
Sequenza di eventi alternativi	-

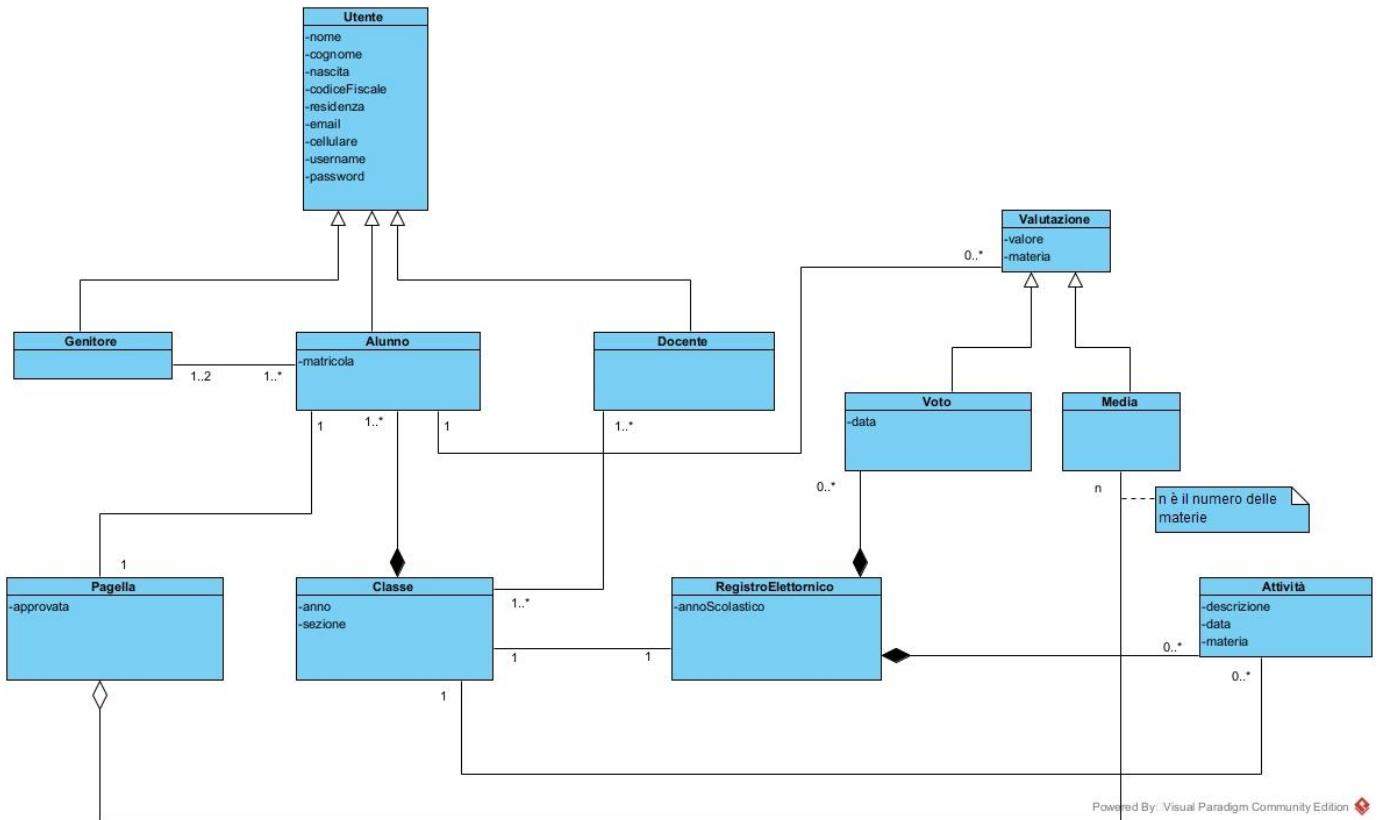
Caso d'uso d'inclusione: Calcolo Media	
Attore primario	-
Attore secondario	-
Descrizione	Il sistema calcola automaticamente la media dei voti assegnati per materia per uno specifico studente.
Pre-Condizioni	Deve essere stato assegnato almeno un voto all'alunno.
Sequenza di eventi	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il sistema lo considera necessario.

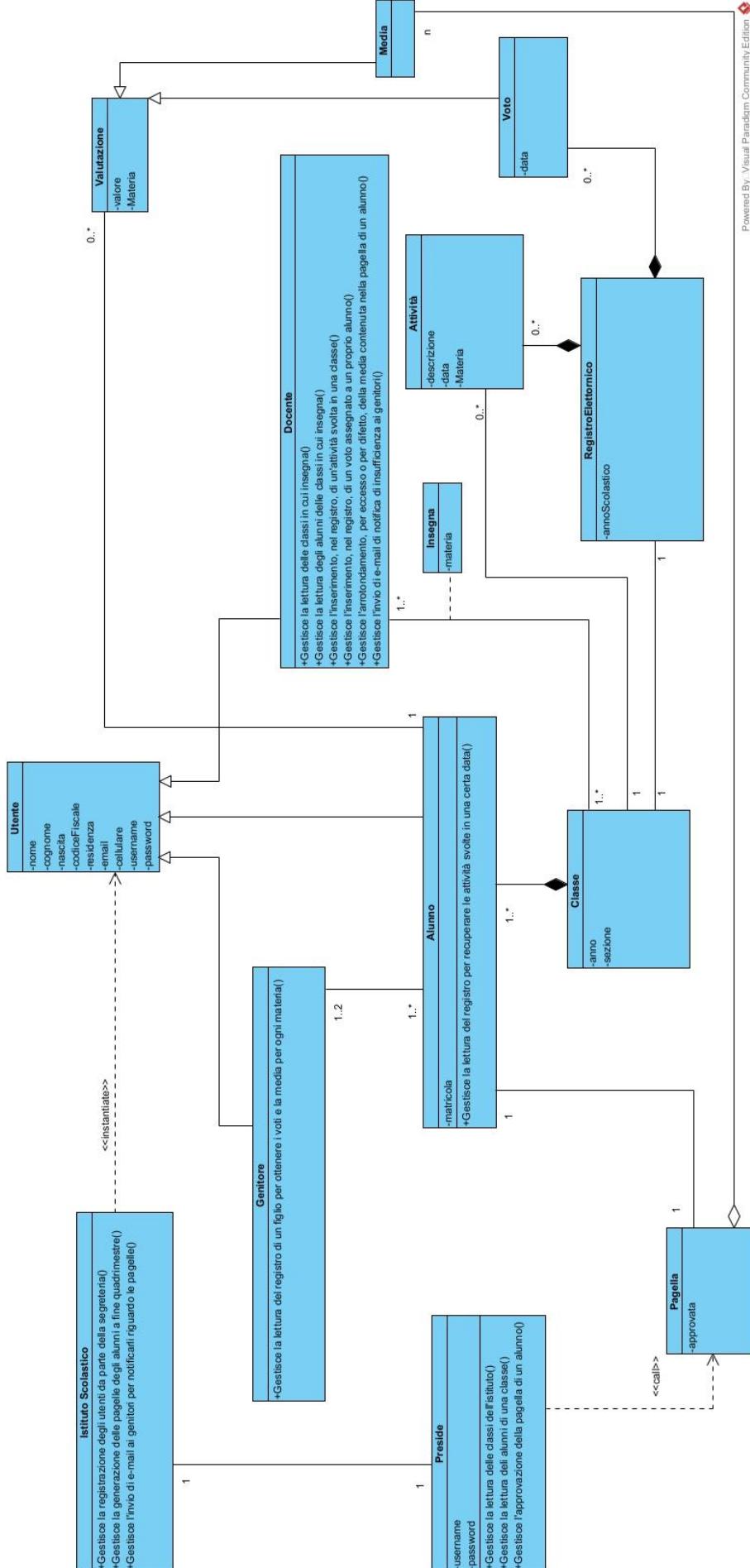
principale	<ol style="list-style-type: none"> 2. FOR ogni materia <ol style="list-style-type: none"> 2.1 Il sistema estrae tutti i voti assegnati all'alunno in questione. 2.2 Il sistema somma i valori prelevati e divide il risultato per il numero di voti.
Post-Condizioni	-
Casi d'uso correlati	<i>Lettura Carriera, Generazione Pagella</i>
Sequenza di eventi alternativi	-

Caso d'uso d'inclusione:	Consulta Classi
Attore primario	Preside, Docente
Attore secondario	-
Descrizione	Un preside o un docente richiede di stampare una lista di classi. Il preside può stampare tutte le classi dell'istituto, il docente solo le classi dove insegna. Per il docente, in aggiunta, vengono mostrate anche le materie insegnate in ogni classe.
Pre-Condizioni	Il docente o il preside deve essere autenticato.
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il sistema lo ritiene necessario. 2. IF l'attore primario è il preside <ol style="list-style-type: none"> 2.1 Il sistema stampa una lista di tutte le classi dell'istituto, spiegandone la sezione, l'anno e anche l'anno scolastico. 3. IF l'attore è un docente <ol style="list-style-type: none"> 3.1. Il sistema stampa una lista di tutte le classi in cui insegna il docente, compresa delle materie insegnate.
Post-Condizioni	-
Casi d'uso correlati	<i>Modifica Pagella, Inserisci Attività, Inserisci Voto, Approvazione Pagella</i>
Sequenza di eventi alternativi	-

Caso d'uso d'inclusione:	Consulta Alunni
Attore primario	Preside, Docente
Attore secondario	-
Descrizione	Un preside o un docente richiede di stampare la lista degli alunni di una classe specificata. Il docente può accedere solo alle classi in cui insegna una materia.
Pre-Condizioni	Il docente o il preside deve essere autenticato.
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il sistema lo ritiene necessario. 2. Il sistema richiede che venga selezionata la classe di cui si desidera conoscere gli alunni. 3. L'attore primario inserisce la sezione e l'anno della classe interessata. 4. Il sistema stampa una lista contenente tutti gli alunni della classe.
Post-Condizioni	-
Casi d'uso correlati	<i>Modifica Pagella, Inserisci Voto, Approvazione Pagella</i>
Sequenza di eventi alternativi	-

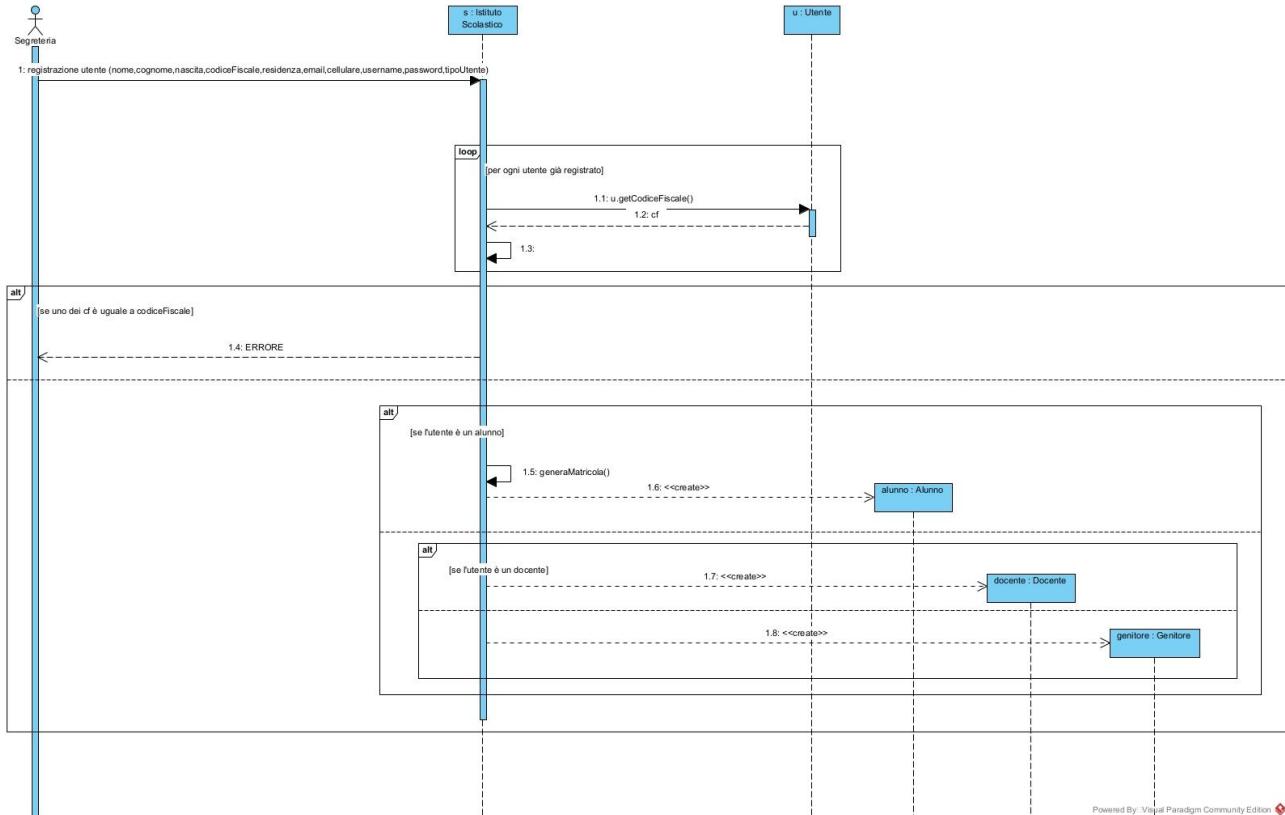
2.6 Diagramma delle classi di analisi



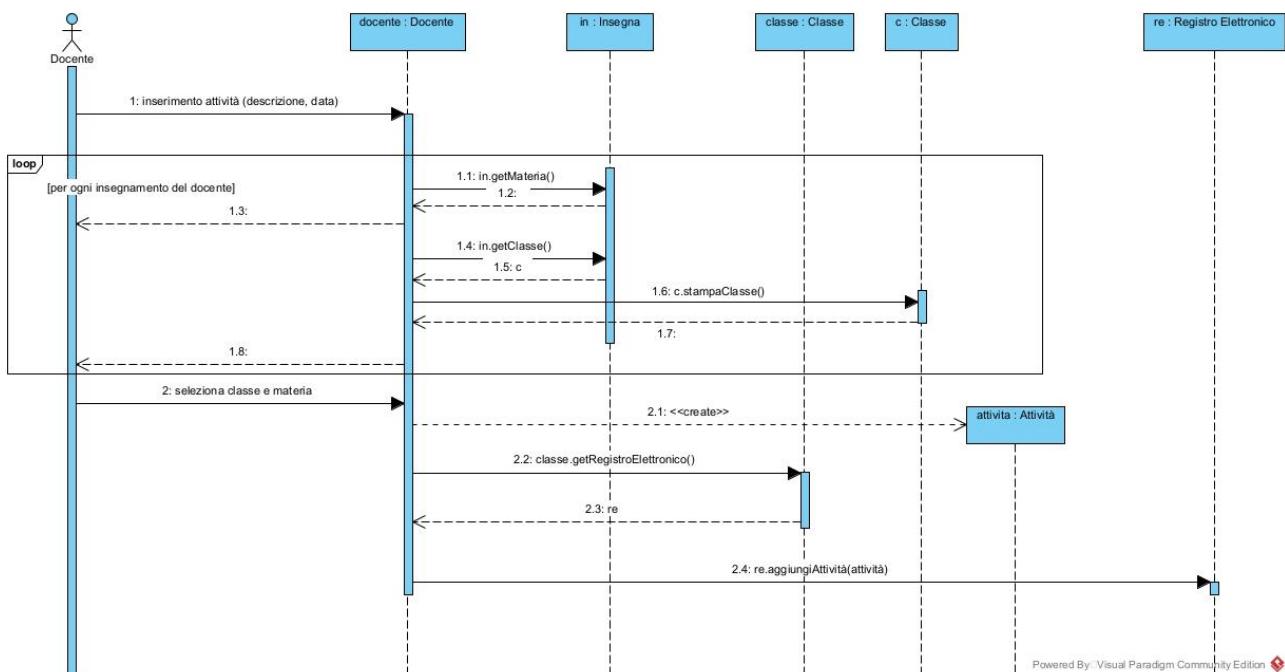


2.7 Diagrammi di sequenza di analisi

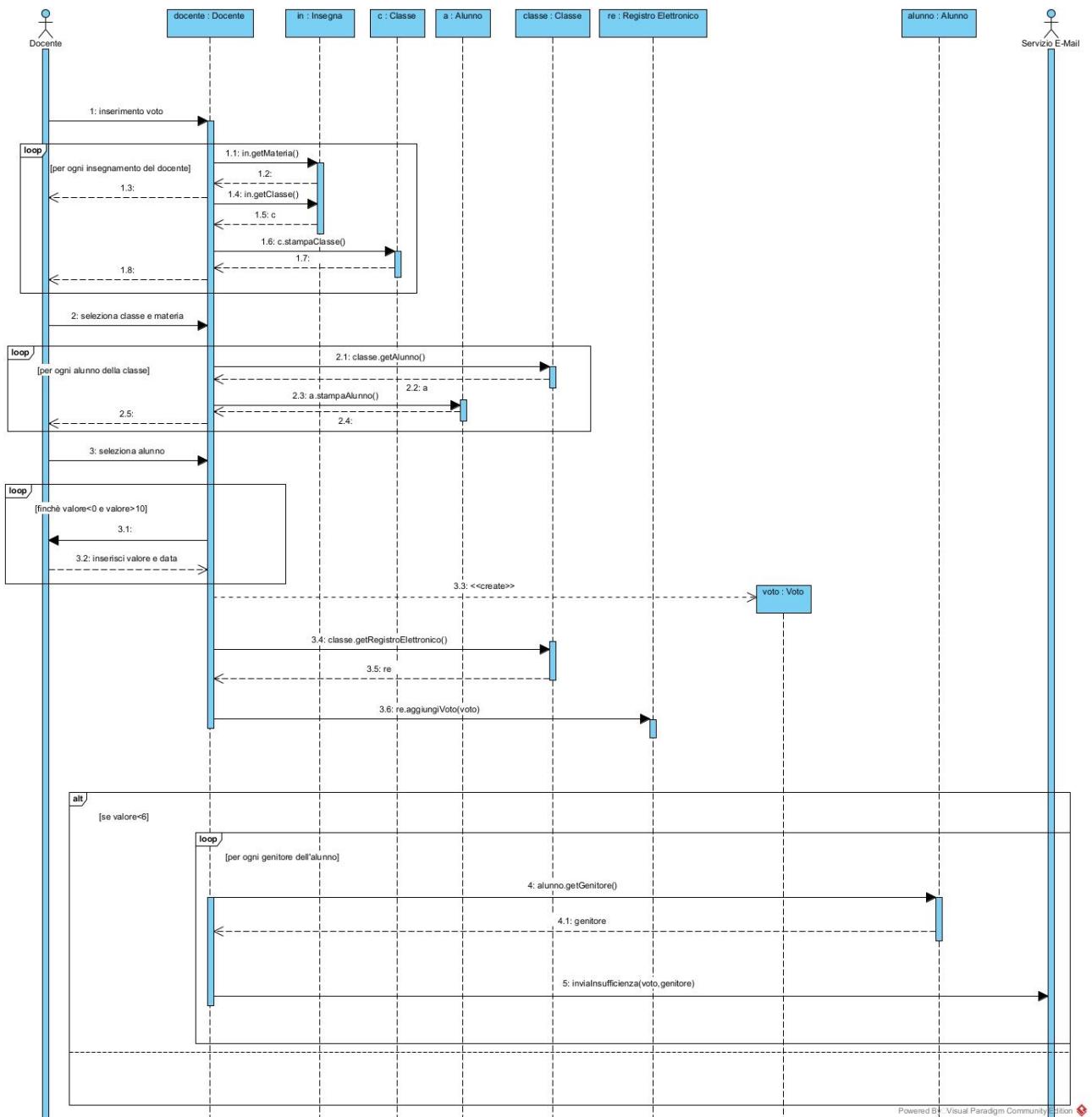
- SD1: Registrazione Utente



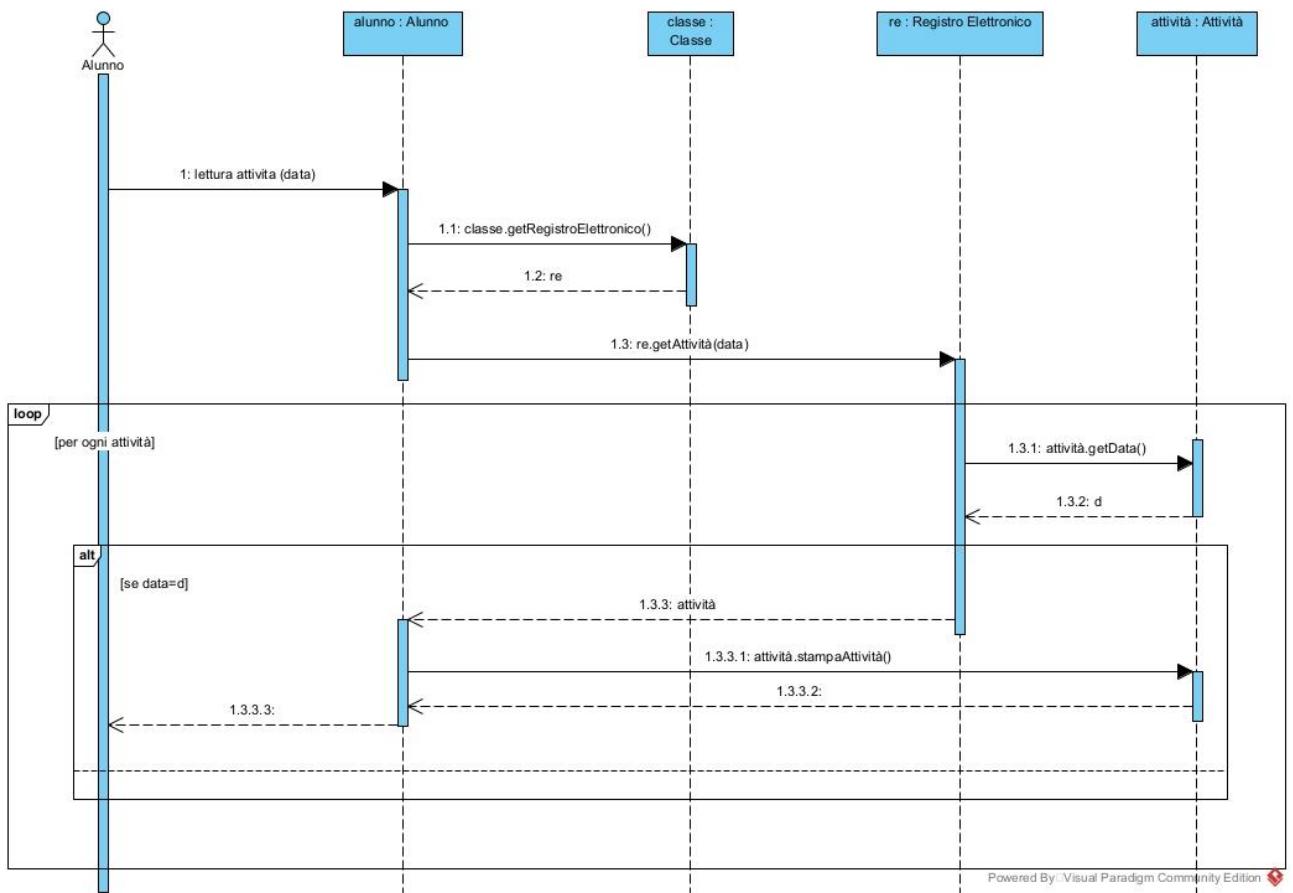
- SD2: Inserimento Attività e Consulta Classi



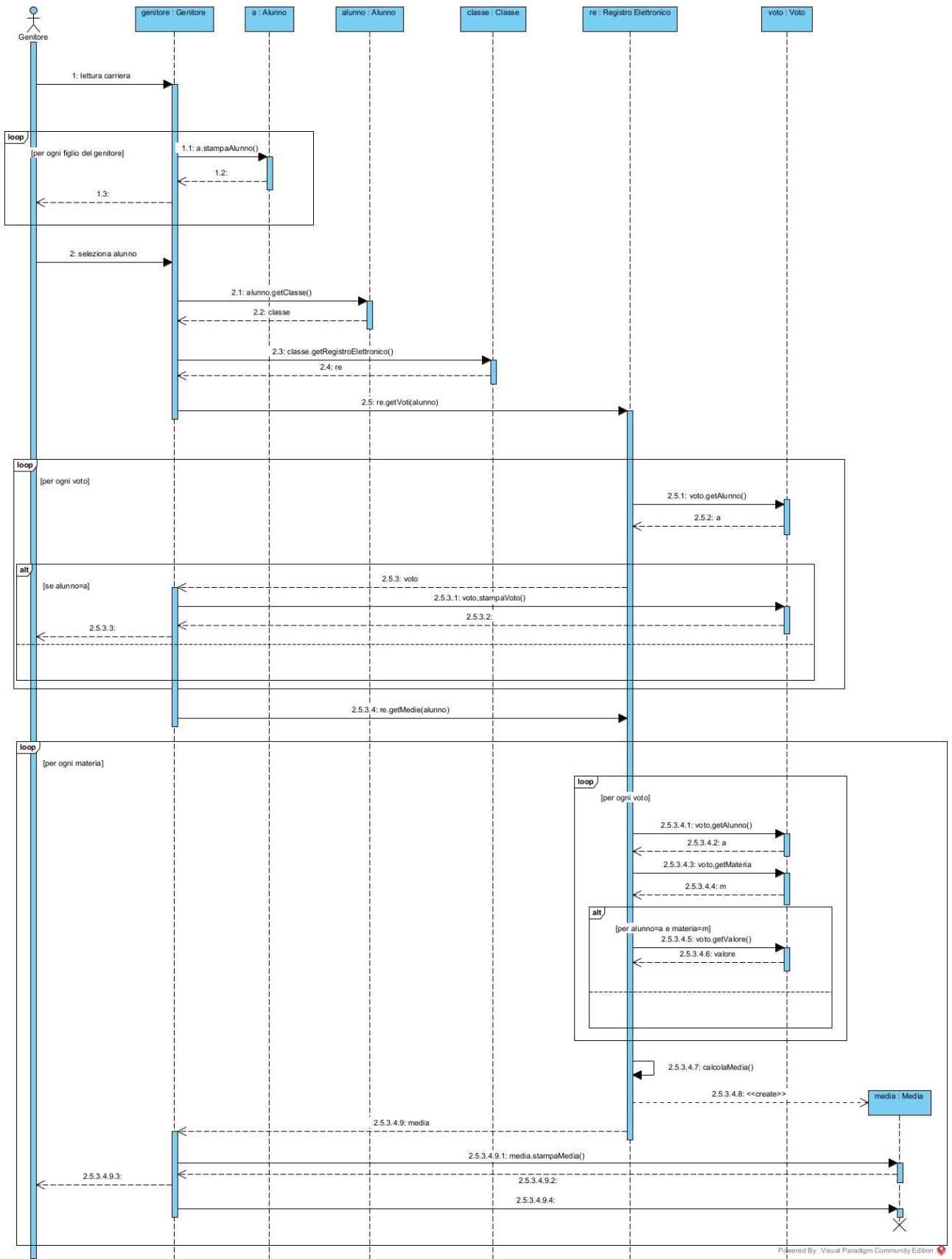
- SD3: Inserimento Voto, Invio E-Mail Insufficienza, Consulta Classi e Consulta Alunni



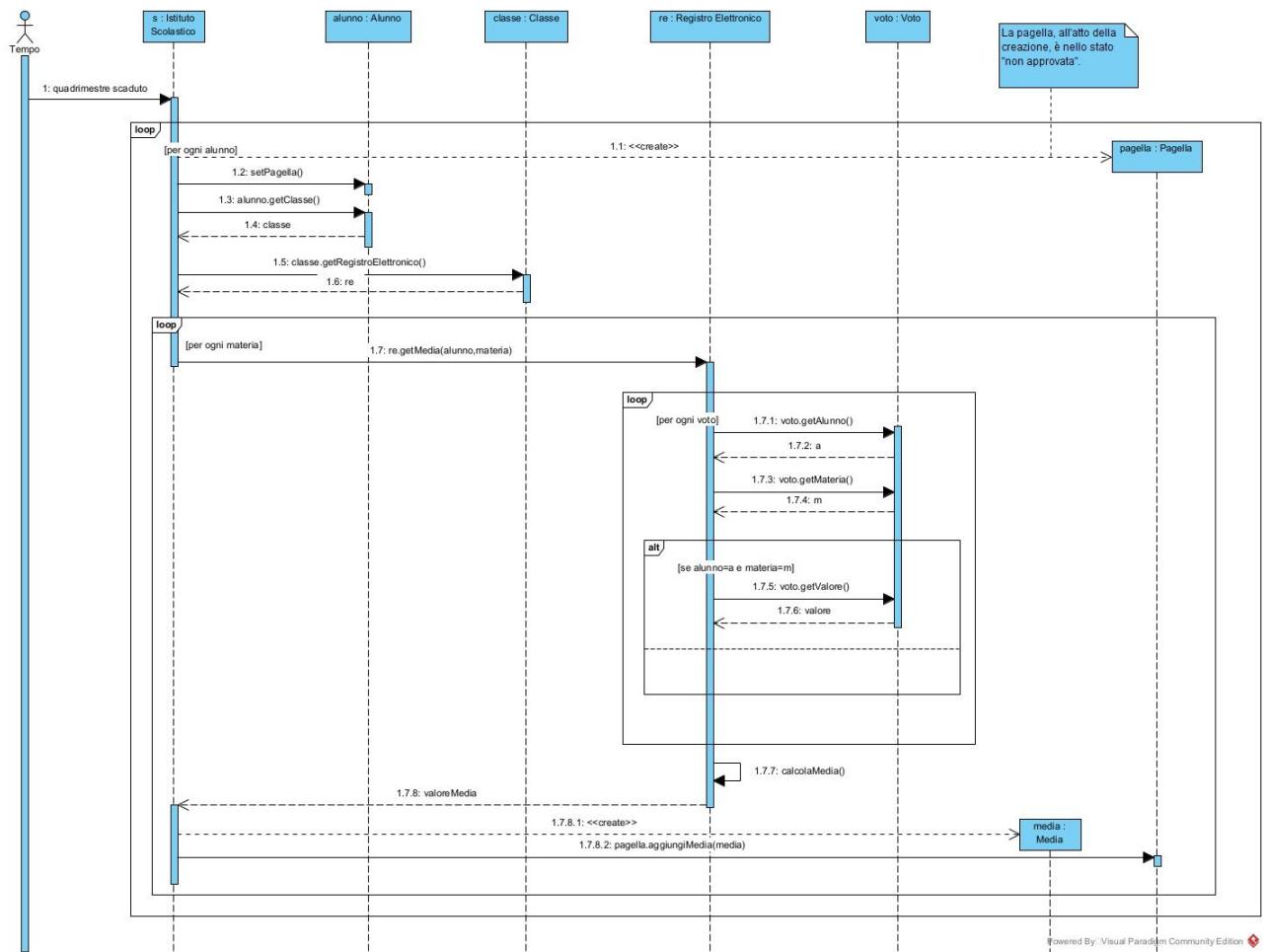
- SD4: Lettura Attività



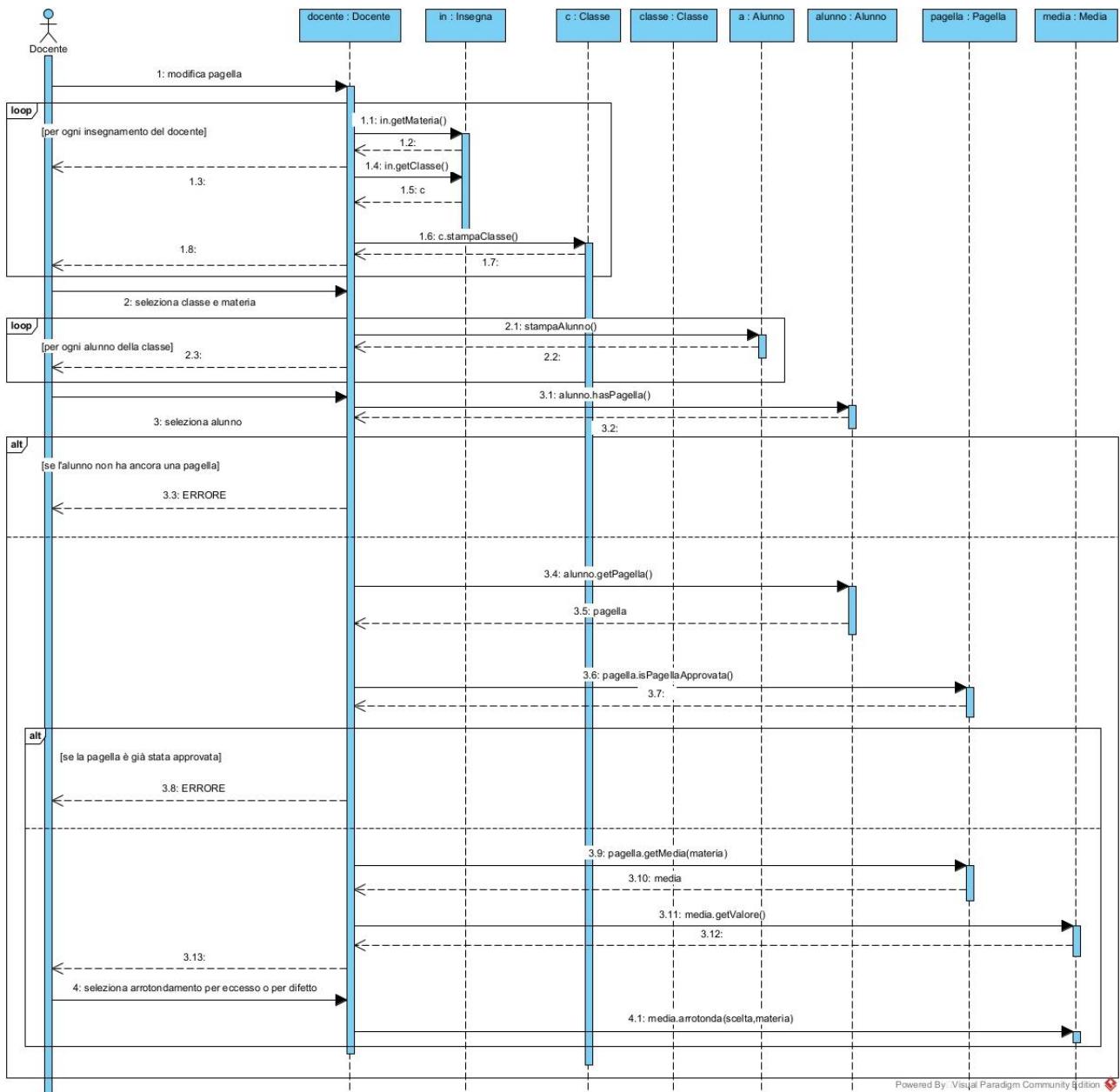
- SD5: Lettura Carriera e Calcolo Media



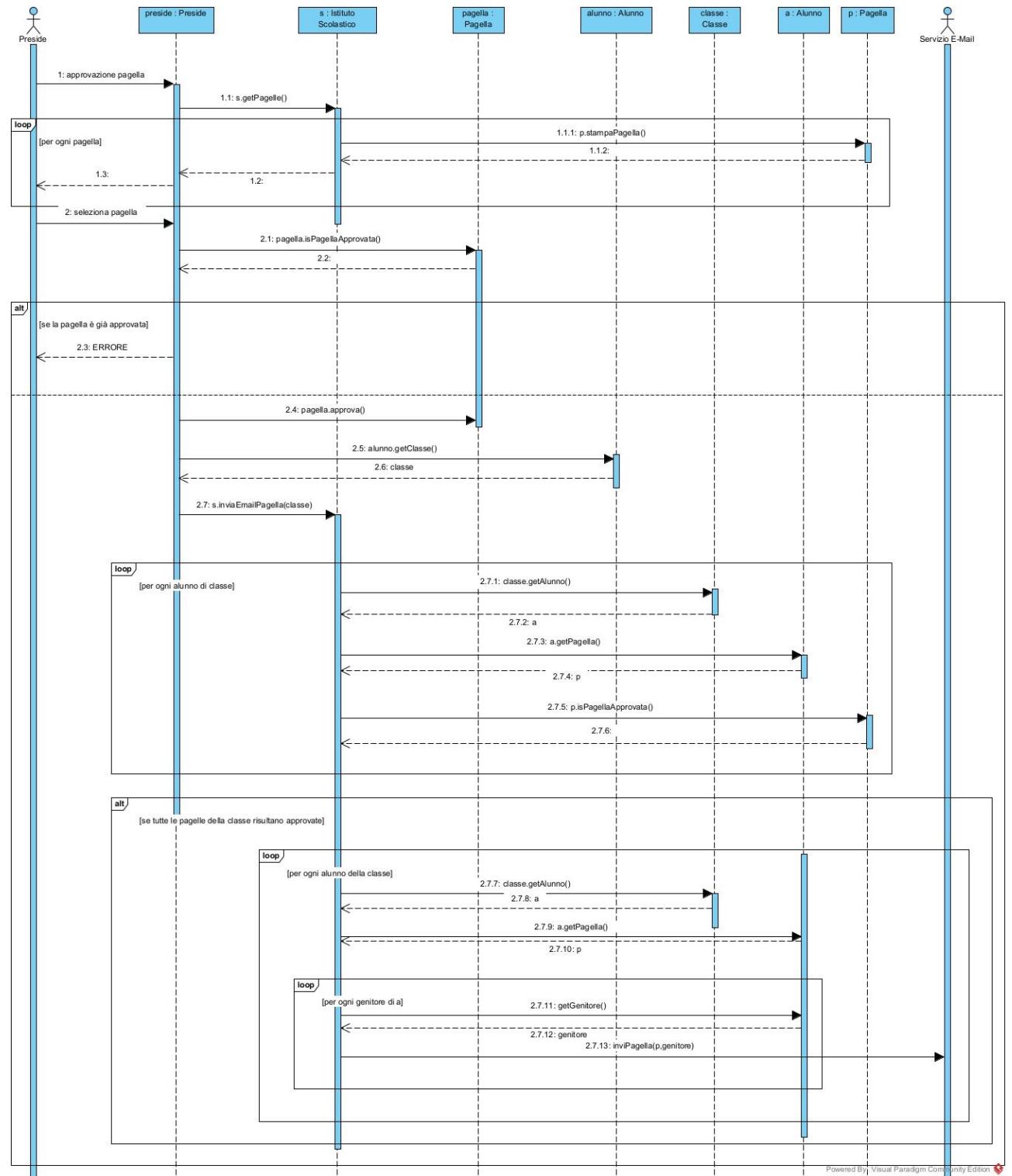
- SD6: Generazione Pagelle e Calcolo Media



- SD7: Modifica Pagella, Consulta Classi e Consulta Alunni



- SD8: Approvazione Pagella, Invio Insufficienza Pagella, Consulta Classi e Consulta Alunni



Powered By Visual Paradigm Community Edition

2.8 Verifica della completezza dei requisiti

Legenda: UCD = Use Case Diagram, CD = Class Diagram, SD = Sequence Diagram

- **RF01** è modellato nell'UCD con l'attore "Segreteria" e con il caso d'uso UC1, nel CD come la classe "Istituto Scolastico" e con SD1.
- **RF02** è modellato nel CD con le classi "Registro Elettronico" e "Classe".
- **RF03** è modellato nell'UCD con l'attore "Preside" e con il caso d'uso UC13 e UC14, e con SD8.
- **RF04** è modellato nell'UCD con l'attore "Docente" e con il caso d'uso UC13 e UC14, e con SD2, SD3 e SD7.
- **RF05** è modellato nell'CD con le classi "Docente", "Classe" e "Insegna".
- **RF06** è modellato nell'UCD con l'attore "Docente" e con il caso d'uso UC3, e con SD2.
- **RF07** è modellato nell'UCD con l'attore "Docente" e con il caso d'uso UC4, e con SD3.
- **RF08** è modellato nell'UCD con l'attore "Servizio E-Mail" e con il caso d'uso UC11, e con SD3.
- **RF09** è modellato nell'UCD con l'attore "Alunno" e il caso d'uso UC5, e con SD4.
- **RF10** è modellato nell'UCD con l'attore "Genitore" e il caso d'uso UC6, e con SD5.
- **RF11** è modellato nell'UCD con l'attore "Tempo" e il caso d'uso UC7, e con SD6.
- **RF12** è modellato nell'UCD con l'attore "Docente" e il caso d'uso UC8, e con SD7.
- **RF13** è modellato nell'UCD con l'attore "Preside" e il caso d'uso UC9, e con SD8.
- **RF14** è modellato nell'UCD con gli attori "Servizio E-Mail" e con il caso d'uso UC10, e con SD8.
- **RD01** è modellato nel CD con le classi "Utente", "Docente", "Alunno" e "Genitore".
- **RD02** è modellato nel CD con gli attributi della classe "Utente" ovvero "nome", "cognome", "nascita", "codiceFiscale", "residenza", "email", "cellulare", "username", "password".
- **RD03** è modellato nel CD con la classe "Classe".
- **RD04** è modellato nel CD con gli attributi "anno" e "sezione" della classe "Classe".
- **RD05** è modellato nel CD con l'attributo "matricola" di "Alunno" e con la classe "Classe".

3. Stima dei costi

Si effettua la stima dei costi secondo la *Function Point Analysis*. È riportata, in seguito, una tabella di riferimento per le complessità dei dati e delle transazioni e una tabella che elenca i fattori correttivi, il cui valore deve essere compreso tra 0 e 5.

	SEMPLICE	MEDIO	COMPLESSO
NILF	7	10	15
NEIF	5	7	10
NEI	3	4	6
NEO	4	5	7
NEQ	4	4	6

FATTORI CORRETTIVI

COMUNICAZIONE DATI

DISTRIBUZIONE ELABORAZIONE

PRESTAZIONI

UTILIZZO INTENSIVO CONFIGURAZIONE

FREQUENZA DELLE TRANSAZIONI

INSERIMENTO DATI INTERATTIVO

EFFICIENZA PER L'UTENTE FINALE

AGGIORNAMENTO INTERATTIVO

COMPLESSITA' ELABORATIVA

RIUSABILITA'

FACILITA' INSTALLAZIONE

FACILITA' GESTIONE OPERATIVA

MOLTEPLICITA' DI SITI

FACILITA' DI MODIFICA

REGISTRAZIONE UTENTE

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
NILF	1	7			7
NEIF	0				0
NEI	9	3			18
NEO	0				0
NEQ	0				0

NILF: Viene registrato un nuovo utente nel sistema.

NEI: La segreteria deve inserire nel sistema il nome, il cognome, la data di nascita, il codice fiscale, il comune di residenza, l'e-mail, il numero di cellulare, l'username e la password dell'utente.

FATTORI CORRETTIVI

UFP= 25
LLOC/UFP = 1.325

COMUNICAZIONE DATI	2
DISTRIBUZIONE ELABORAZIONE	0
PRESTAZIONI	2
UTILIZZO INTENSIVO CONFIGURAZIONE	0
FREQUENZA DELLE TRANSAZIONI	2
INSERIMENTO DATI INTERATTIVO	0
EFFICIENZA PER L'UTENTE FINALE	3
AGGIORNAMENTO INTERATTIVO	2
COMPLESSITÀ ELABORATIVA	0
RIUSABILITÀ	4
FACILITÀ INSTALLAZIONE	2
FACILITÀ GESTIONE OPERATIVA	2
Molteplicità DI SITI	0
FACILITÀ DI MODIFICA	3
	22

FP= 21,7
JAVA = 1.153

LETTURA CARRIERA

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
NILF	1		10		10
NEIF	0				0
NEI	1	3			3
NEO	1		5		5
NEQ	1	4			4

NILF: Il sistema deve accedere al registro elettronico della classe dell'alunno.

NEI: Il genitore deve specificare l'alunno figlio.

NEO: Il sistema deve calcolare e restituire le medie per ogni materia a partire dai voti.

NEQ: Il sistema restituire i voti assegnati all'alunno.

FATTORI CORRETTIVI

COMUNICAZIONE DATI	3	UFP= 22
DISTRIBUZIONE ELABORAZIONE	2	LLOC/UFP = 1.166
PRESTAZIONI	3	
UTILIZZO INTENSIVO CONFIGURAZIONE	0	
FREQUENZA DELLE TRANSAZIONI	2	
INSERIMENTO DATI INTERATTIVO	0	
EFFICIENZA PER L'UTENTE FINALE	3	
AGGIORNAMENTO INTERATTIVO	0	FP= 19.1
COMPLESSITÀ ELABORATIVA	3	JAVA = 995
RIUSABILITÀ	1	
FACILITÀ INSTALLAZIONE	1	
FACILITÀ GESTIONE OPERATIVA	2	
Molteplicità DI SITI	0	
FACILITÀ DI MODIFICA	2	
	22	

INVIO EMAIL PAGELLA

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
NILF	1	7			7
NEIF	1		7		7
NEI	1	3			3
NEO	0				0
NEQ	0				0

NILF: Il sistema deve recuperare l'e-mail del genitore dell'alunno.

NEIF: Il sistema deve comunicare col servizio e-mail per passargli le informazioni necessarie.

NEI: Un'altra parte del sistema ha la responsabilità di passare la pagella di un alunno.

FATTORI CORRETTIVI

UFP= 17
LLOC/UFP = 901

COMUNICAZIONE DATI	3
DISTRIBUZIONE ELABORAZIONE	0
PRESTAZIONI	2
UTILIZZO INTENSIVO CONFIGURAZIONE	1
FREQUENZA DELLE TRANSAZIONI	1
INSERIMENTO DATI INTERATTIVO	0
EFFICIENZA PER L'UTENTE FINALE	1
AGGIORNAMENTO INTERATTIVO	0
COMPLESSITÀ ELABORATIVA	1
RIUSABILITÀ	3
FACILITÀ INSTALLAZIONE	3
FACILITÀ GESTIONE OPERATIVA	3
Molteplicità DI SITI	0
FACILITÀ DI MODIFICA	1
	19

FP= 14,9
JAVA = 757

4. Piano di test funzionale

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

PIANO DI TEST PER LA FUNZIONALITÀ “*Registrazione Utente*”.

NOME, COGNOME, RESIDENZA	NASCITA	CODICEFISCALE	EMAIL	CELLULARE	USERNAME, PASSWORD
<ul style="list-style-type: none">Stringa di caratteri di lunghezza <= 30Stringa di caratteri di lunghezza > 30 [ERROR]Stringa che contiene simboli che non sono caratteri [ERROR]	<ul style="list-style-type: none">Data con formato valido (gg/mm/aaaa)Data con formato non valido [ERROR]	<ul style="list-style-type: none">Stringa di 16 caratteriStringa di caratteri di lunghezza diversa da 16 [ERROR]Stringa che contiene simboli che non sono caratteri [ERROR]	<ul style="list-style-type: none">Stringa in cui è presente il simbolo @Stringa in cui non è presente il simbolo @ [ERROR]	<ul style="list-style-type: none">Numero intero >= 1000000000 e <= 9999999999Numero intero < 1000000000 [ERROR]Numero intero > 9999999999 [ERRORE]	<ul style="list-style-type: none">Stringa di lunghezza <= 30Stringa di lunghezza > 30 [ERROR]

Il numero di test da effettuarsi senza particolari vincoli è $3^*3^*3^*2^*3^*2^*3^*2^*2 = 3888$.

Il numero di test da eseguire per testare singolarmente i vincoli è 14 (2 per Nome, 2 per Cognome, 2 per Residenza, 1 per Nascita, 2 per Codice Fiscale, 1 per Email, 2 per Cellulare, 1 per Username e 1 per Password).

Quindi il numero di test risultante è 15.

TEST SUITE PER “*Registrazione Utente*”

5	Cognome – stringa che contiene simboli che non sono caratteri	Nome valido Cognome [ERROR] Nascita valido Codice Fiscale valido Residenza valida Email valida Cellulare valido Username valida Password valida		{Nome: "Mario", Cognome: "Rossi++*°", Data: 10/11/2002, CodiceFiscale: "RSSNMO02S10F839S", Residenza: "Napoli", Email: "mariorossi@gmail.com", Cellulare: 345678912, Username: "mariorossi", Password: "mariocalcetto12"}	Cognome contenente simboli!	
6	Nascita - formato non valido	Nome valido Cognome valido Nascita [ERROR] Codice Fiscale valido Residenza valida Email valida Cellulare valido Username valida Password valida		{Nome: "Mario", Cognome: "Rossi", Data: dieci/13/2002, CodiceFiscale: "RSSNMO02S10F839S", Residenza: "Napoli", Email: "mariorossi@gmail.com", Cellulare: 345678912, Username: "mariorossi", Password: "mariocalcetto12"}	Data non valida!	
7	Codice Fiscale- stringa di caratteri di lunghezza diversa da 16	Nome valido Cognome valido Nascita valida Codice Fiscale [ERROR] Residenza valida Email valida Cellulare valido Username valida Password valida		{Nome: "Mario", Cognome: "Rossi", Data: 10/11/2002, CodiceFiscale: "RSSNMO02S10F83555559SAS16T", Residenza: "Napoli", Email: "mariorossi@gmail.com", Cellulare: 345678912, Username: "mariorossi", Password: "mariocalcetto12"}	Codice fiscale troppo lungo!	
8	Residenza- stringa di caratteri di lunghezza > 30	Nome valido Cognome valido Nascita valida Codice Fiscale valido		{Nome: "Mario", Cognome: "Rossi", Data: 10/11/2002, CodiceFiscale: "RSSNMO02S10F839S",	Residenza troppo lunga!	

	< 1000000000 [ERROR]	Codice Fiscale valido Residenza valida Email valida Cellulare [ERROR] Username valida Password valida		CodiceFiscale: "RSSNMO02S10F839S", Residenza: "Napoli", Email: "mariorossi@gmail.com", Cellulare: 342, Username: "mariorossi", Password: "mariocalcetto12"}		
12	Cellulare - numero intero > 9999999999 [ERROR]	Nome valido Cognome valido Nascita valida Codice Fiscale valido Residenza valida Email valida Cellulare [ERROR] Username valida Password valida		{Nome: "Mario", Cognome: "Rossi", Data: 10/11/2002, CodiceFiscale: "RSSNMO02S10F839S", Residenza: "Napoli", Email: "mariorossi@gmail.com", Cellulare: 3429769769763838, Username: "mariorossi", Password: "mariocalcetto12"}	Cellulare invalido!	
13	Username - stringa di caratteri di lunghezza >30	Nome valido Cognome valido Nascita valido Codice Fiscale valido Residenza valida Email valida Cellulare valido Username [ERROR] Password valida		{Nome: "Mario", Cognome: "Rossi", Data: 10/11/2002, CodiceFiscale: "RSSNMO02S10F839S", Residenza: "Napoli", Email: "mariorossi@gmail.com", Cellulare: 345678912, Username: "mariorossiaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaa", Password: "mariocalcetto12"}	Username troppo lungo!	
14	Password - stringa di caratteri di lunghezza > 30	Nome valido Cognome valido Nascita valido Codice Fiscale valido		{Nome: "Mario", Cognome: "Rossi", Data: 10/11/2002, CodiceFiscale: "RSSNMO02S10F839S",	Password troppo lunga!	

PIANO DI TEST PER LA FUNZIONALITÀ “*Inserimento Voto*”.

DATA	VALORE VOTO
<ul style="list-style-type: none">• Data con formato valido(gg/mm/aaaa)• Data con formato non valido [ERROR]	<ul style="list-style-type: none">• Numero ≥ 0 e ≤ 10• Numero intero < 0 [ERROR]• Numero > 10 [ERROR]

Il numero di test da effettuarsi senza particolari vincoli è $2^*3 = 6$.

Il numero di test da eseguire per testare singolarmente i vincoli è 3 (1 per Data e 2 per Valore Voto).

Quindi il numero di test risultante è 4.

TEST SUITE PER “Inserimento Voto”

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Tutti input validi	Data valido Valore valido		{Data: "20/05/2022", Valore: 7}	Voto inserito correttamente	È stato inserito un nuovo voto nel sistema.

2	Data- formato non valido	Data [ERROR] Valore valido		{Data: "xx/100/3022", Valore: 7}	Data non corretta!	Non è stato aggiunto nessun voto nel sistema.
3	Valore – numero intero < 0	Data valido Valore [ERROR]		{Data: "20/05/2022", Valore: -3}	Valore non valido!	
4	Valore – numero intero > 10	Data valido Valore [ERROR]		{Data: "20/05/2022", Valore: 34}	Valore non valido!	

PIANO DI TEST PER LA FUNZIONALITÀ “Inserimento Attività”.

DESCRIZIONE	DATA
<ul style="list-style-type: none"> Stringa di caratteri di lunghezza <= 30 Stringa di caratteri di lunghezza > 30 [ERROR] Stringa che contiene simboli che non sono caratteri [ERROR] 	<ul style="list-style-type: none"> Data con formato valido(gg/mm/aaaa) Data con formato non valido [ERROR]

Il numero di test da effettuarsi senza particolari vincoli è $3^2 = 9$.

Il numero di test da eseguire per testare singolarmente i vincoli è 3 (2 per Descrizione e 1 per Data).
Quindi il numero di test risultante è 4.

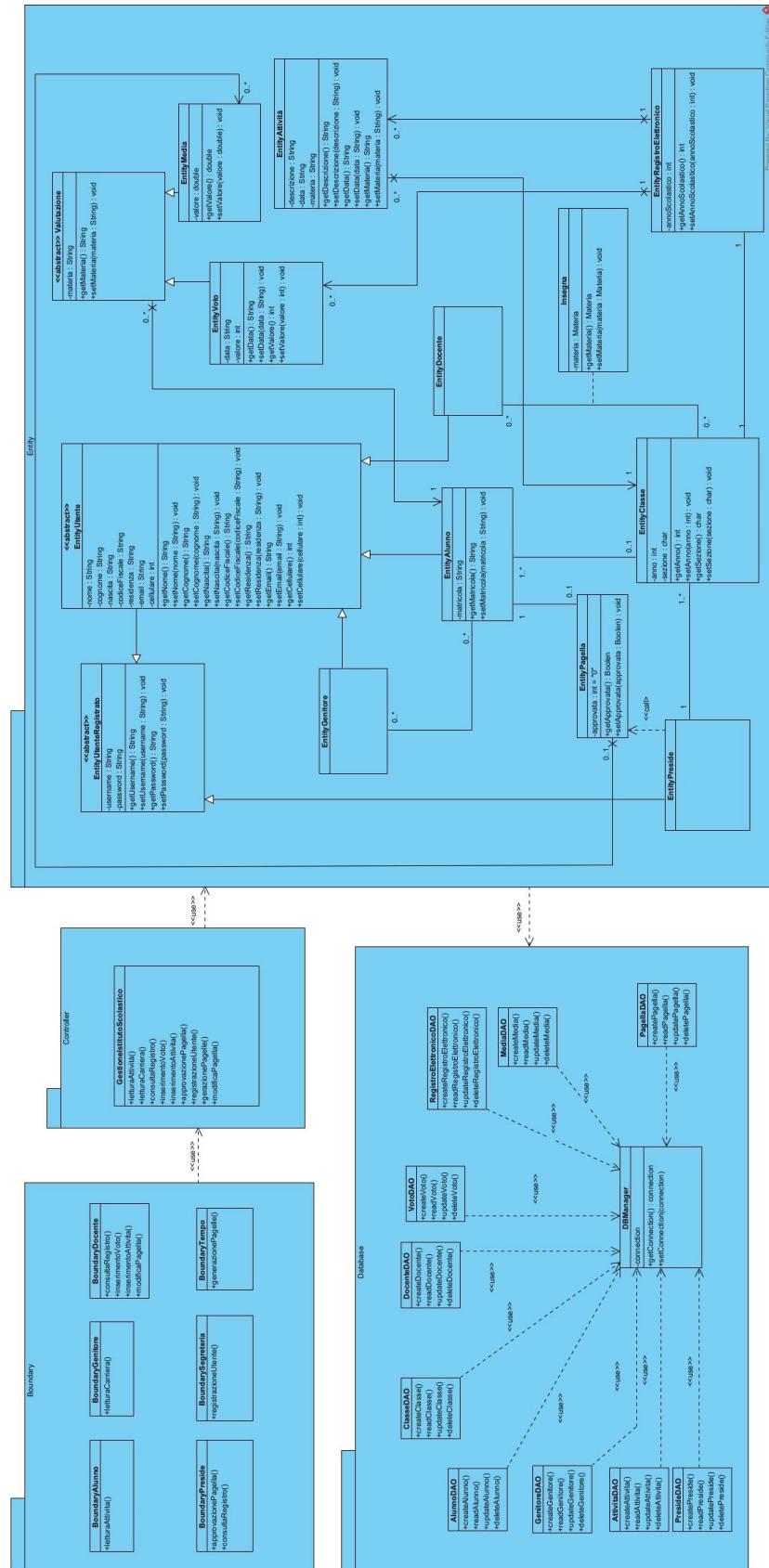
TEST SUITE PER “Inserimento Attività”

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Tutti input validi	Data valida Descrizione valida		{Data: "20/05/2022", Descrizione: "Lettura Dante"}	Attività inserita correttamente!	È stato inserito una

						nuova attività nel sistema.
2	Descrizione - stringa di caratteri di lunghezza > 30	Data valida Descrizione [ERROR]		{Data: "20/05/2022", Descrizione: "Letturaaaaaaaaaaa Danteeeeeeeeeee ooooooooooooooooooooooooooooooo"} oooooooooooooooo	Descrizione troppo lunga!	Non è stata aggiunta nessuna attività nel sistema.
3	Descrizione - stringa che contiene simboli che non sono caratteri	Data valida Descrizione [ERROR]		{Data: "20/05/2022", Descrizione: "Le\tt ura çDant§e!"}	Descrizione errata!	
4	Data - formato non valido	Data [ERRORE] Descrizione valida		{Data: "venti/25/3322", Descrizione: "Lettura Dante"}	Data non valida!	

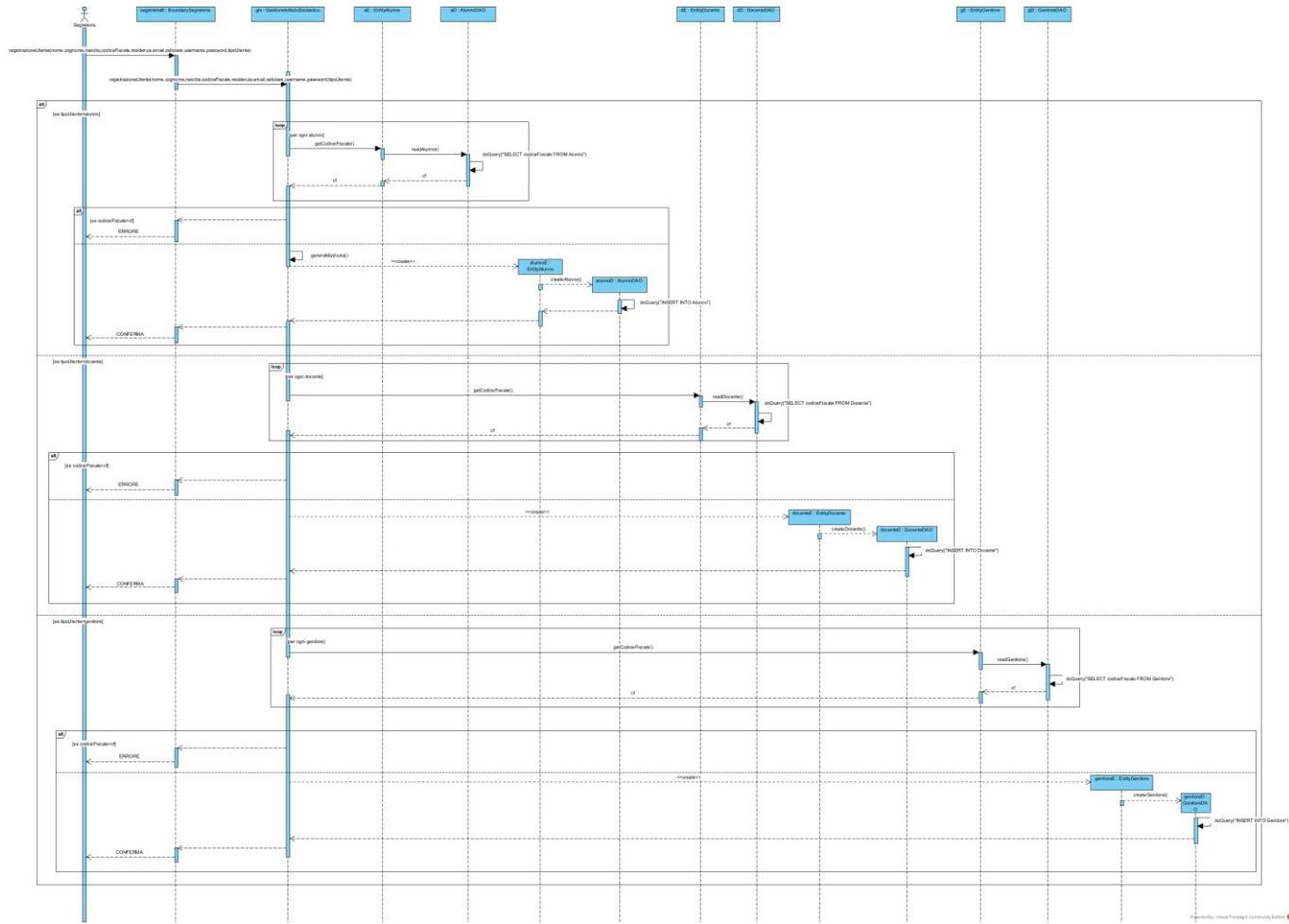
5. Progettazione

5.1 Diagramma delle classi

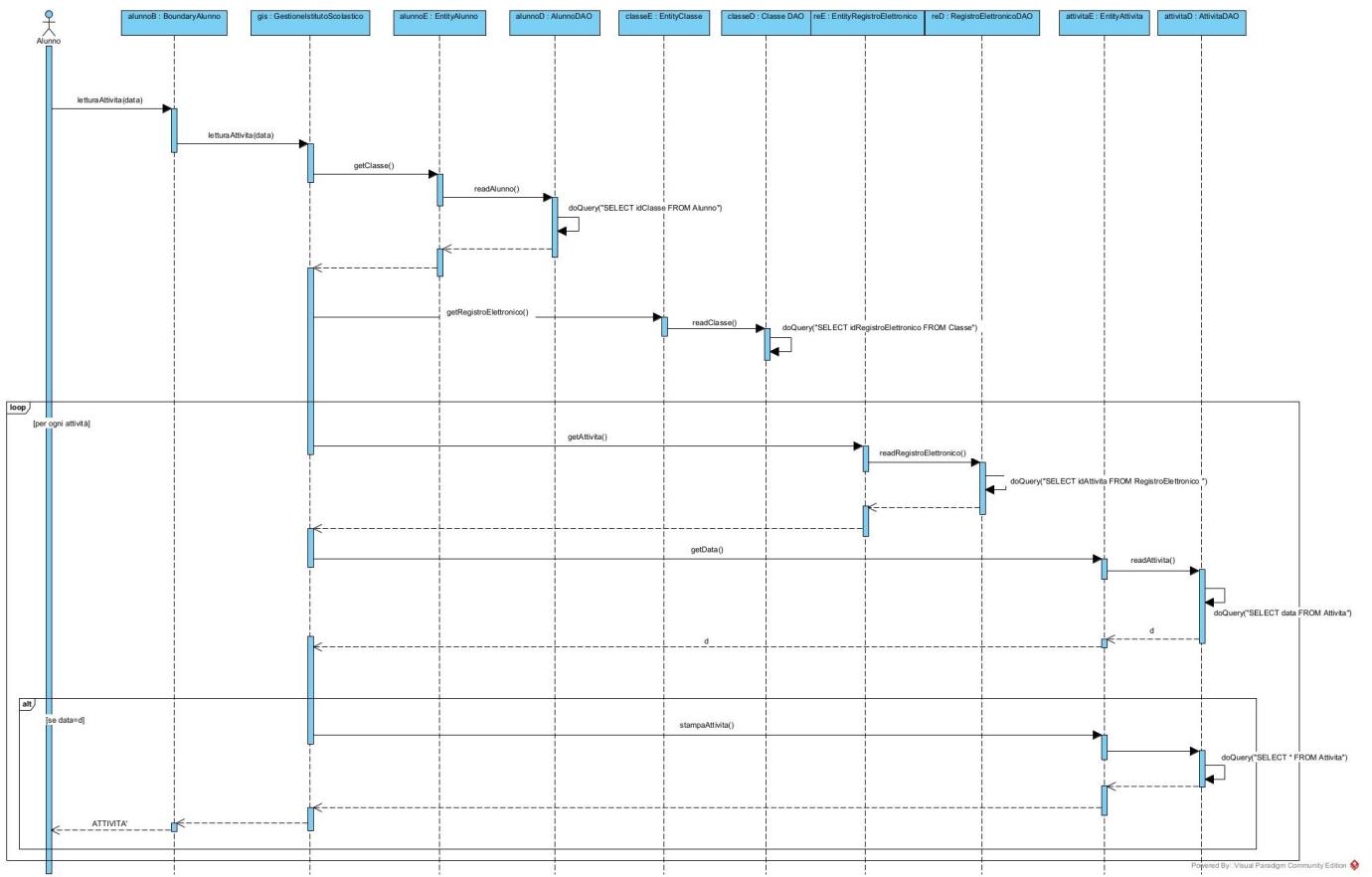


5.2 Diagrammi di sequenza

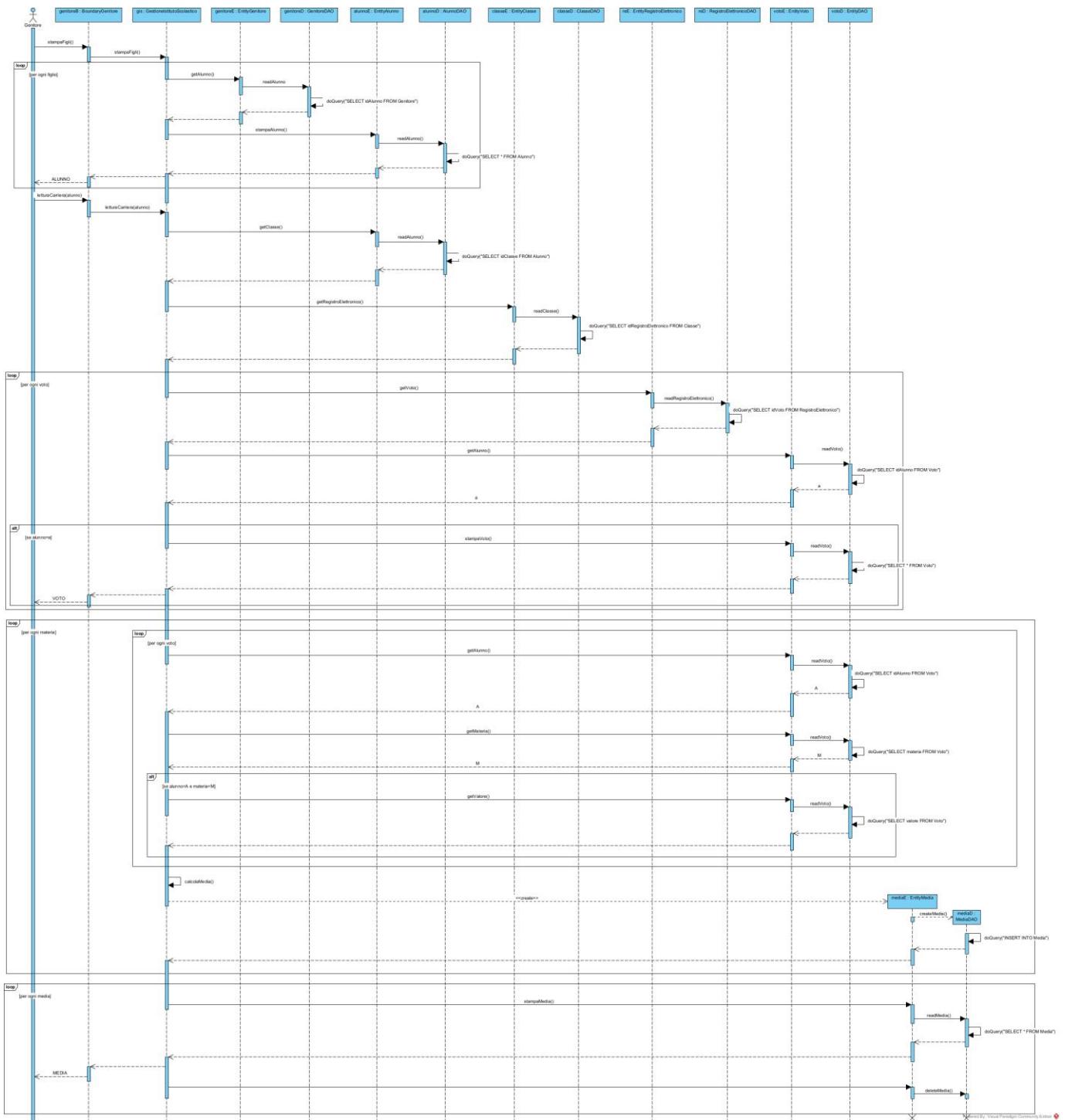
- Registrazione Utente



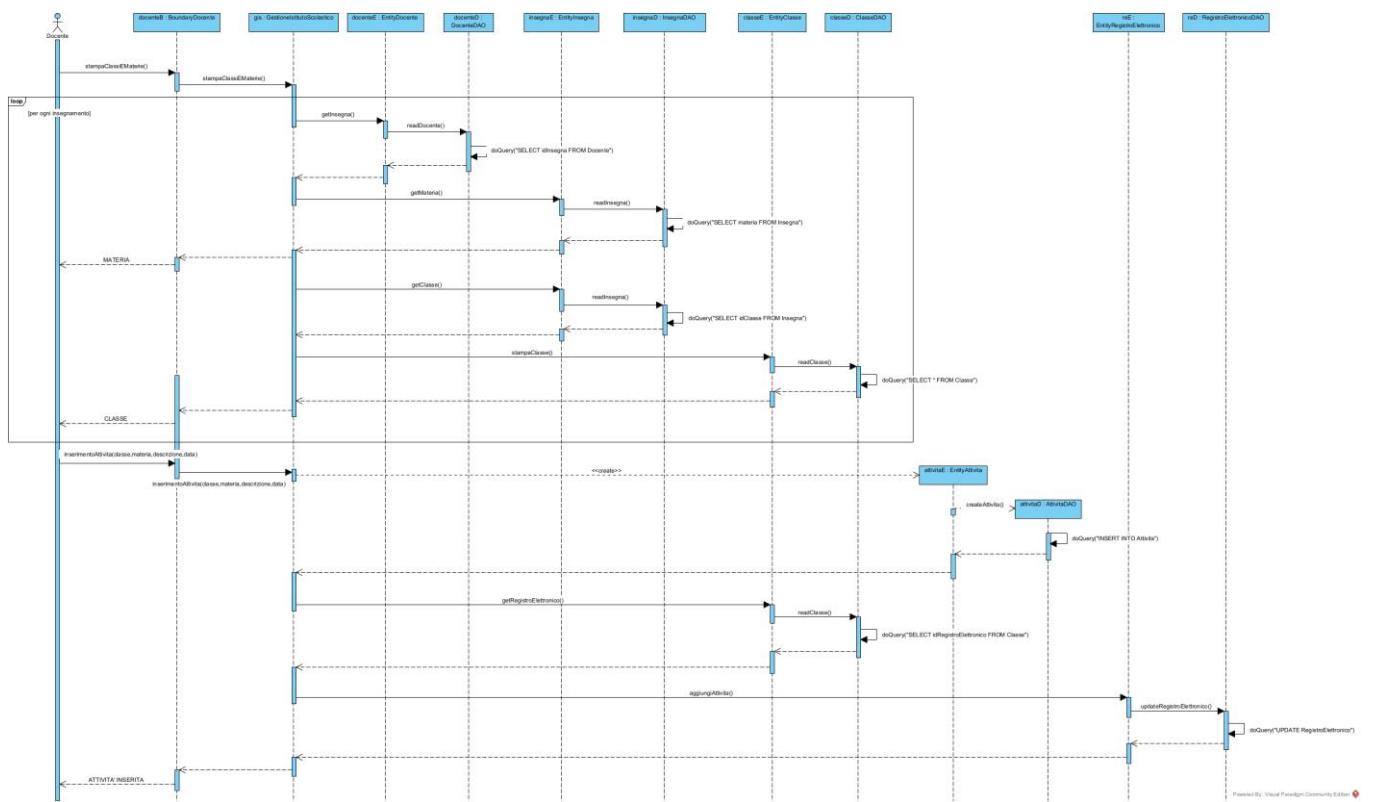
- Lettura Attività



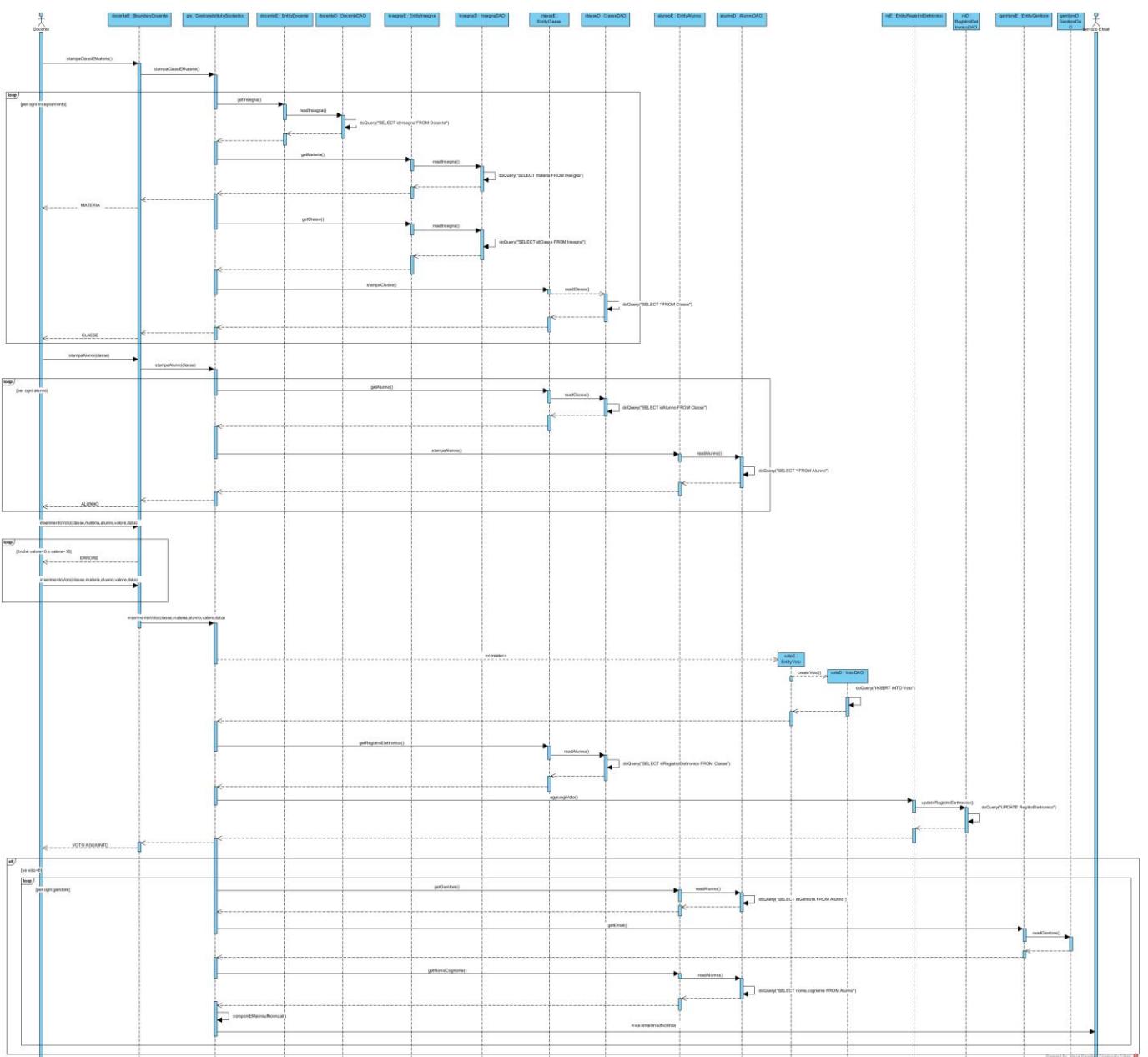
• Lettura Carriera



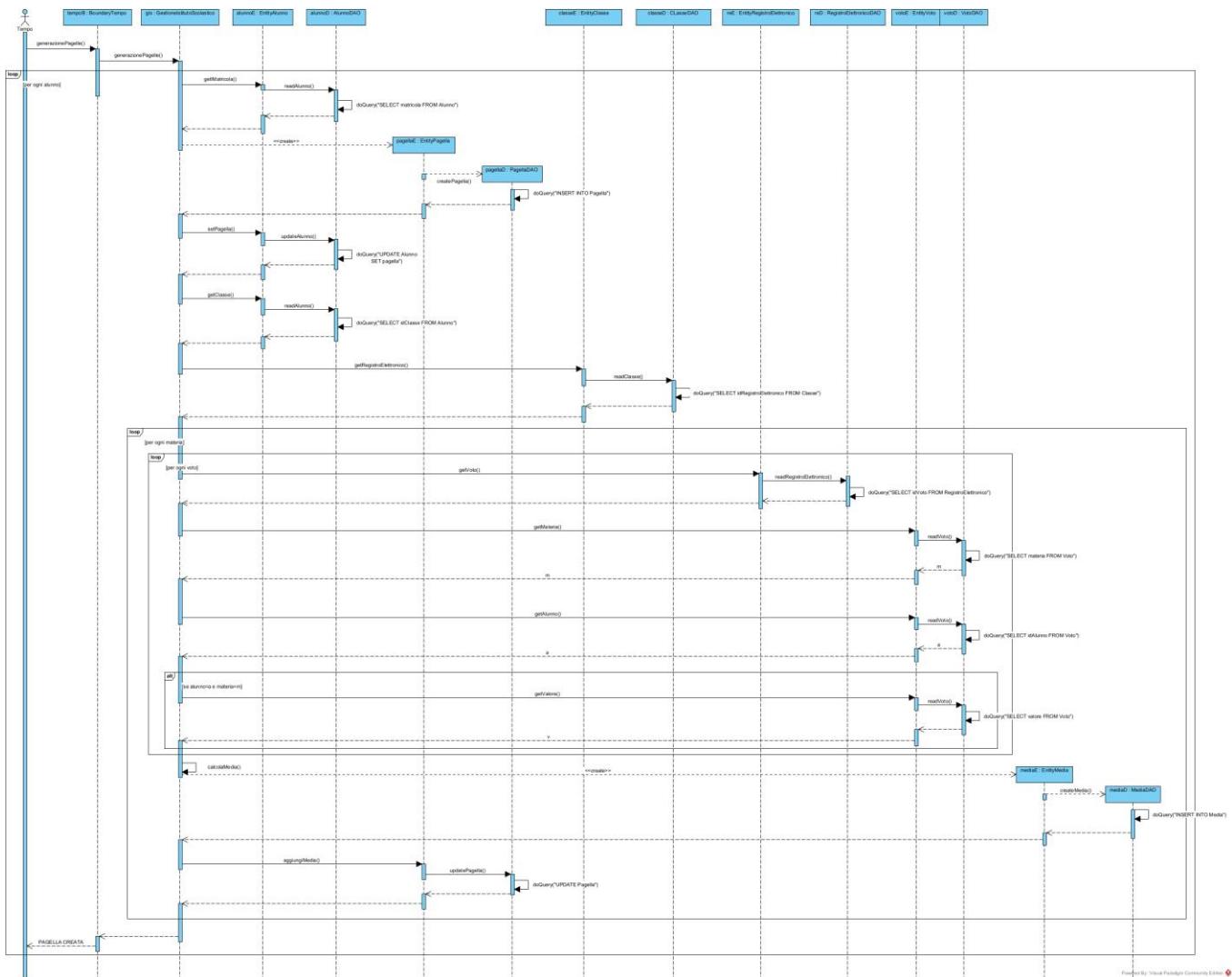
- Inserimento Attività



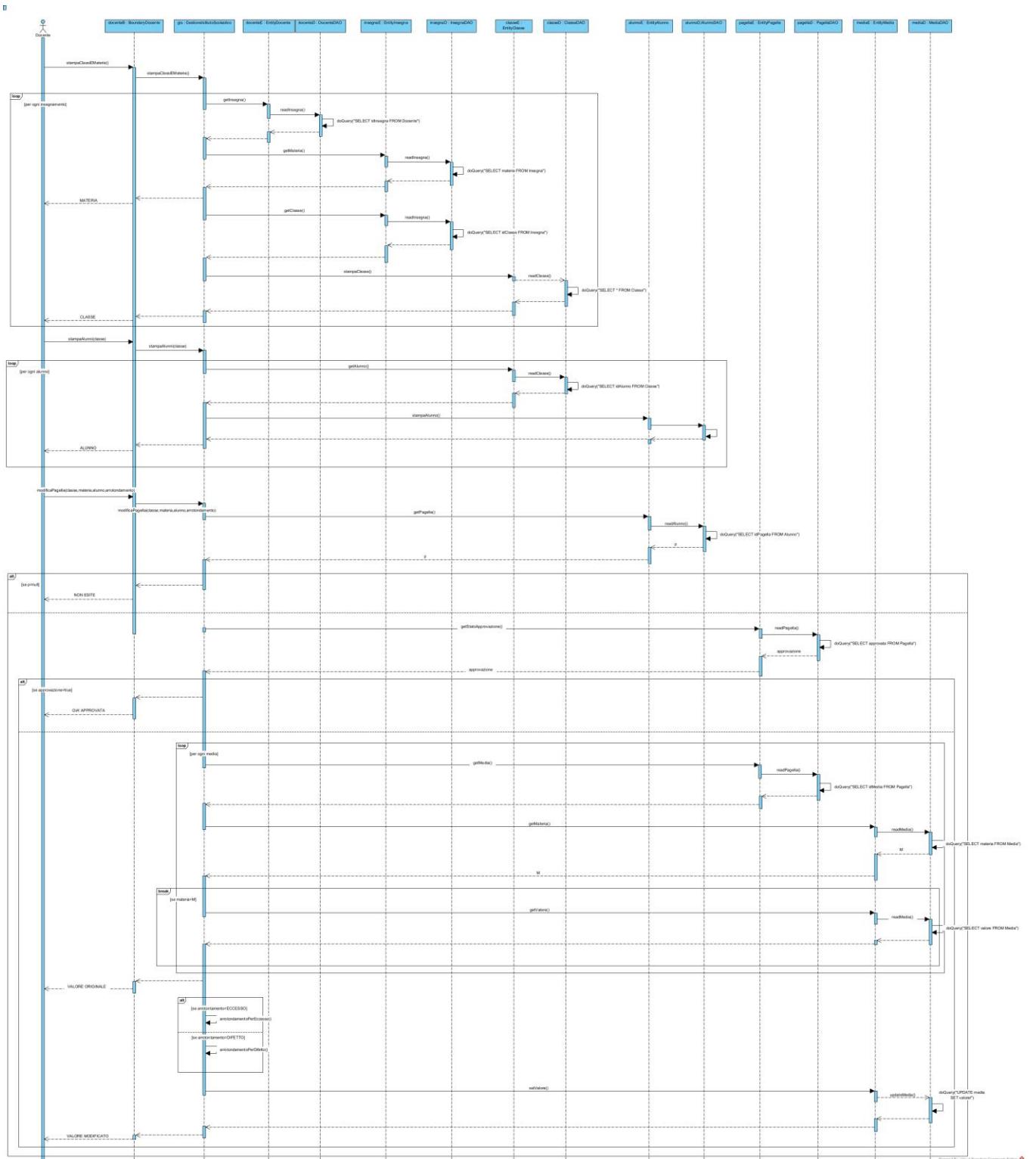
- Inserimento Voto



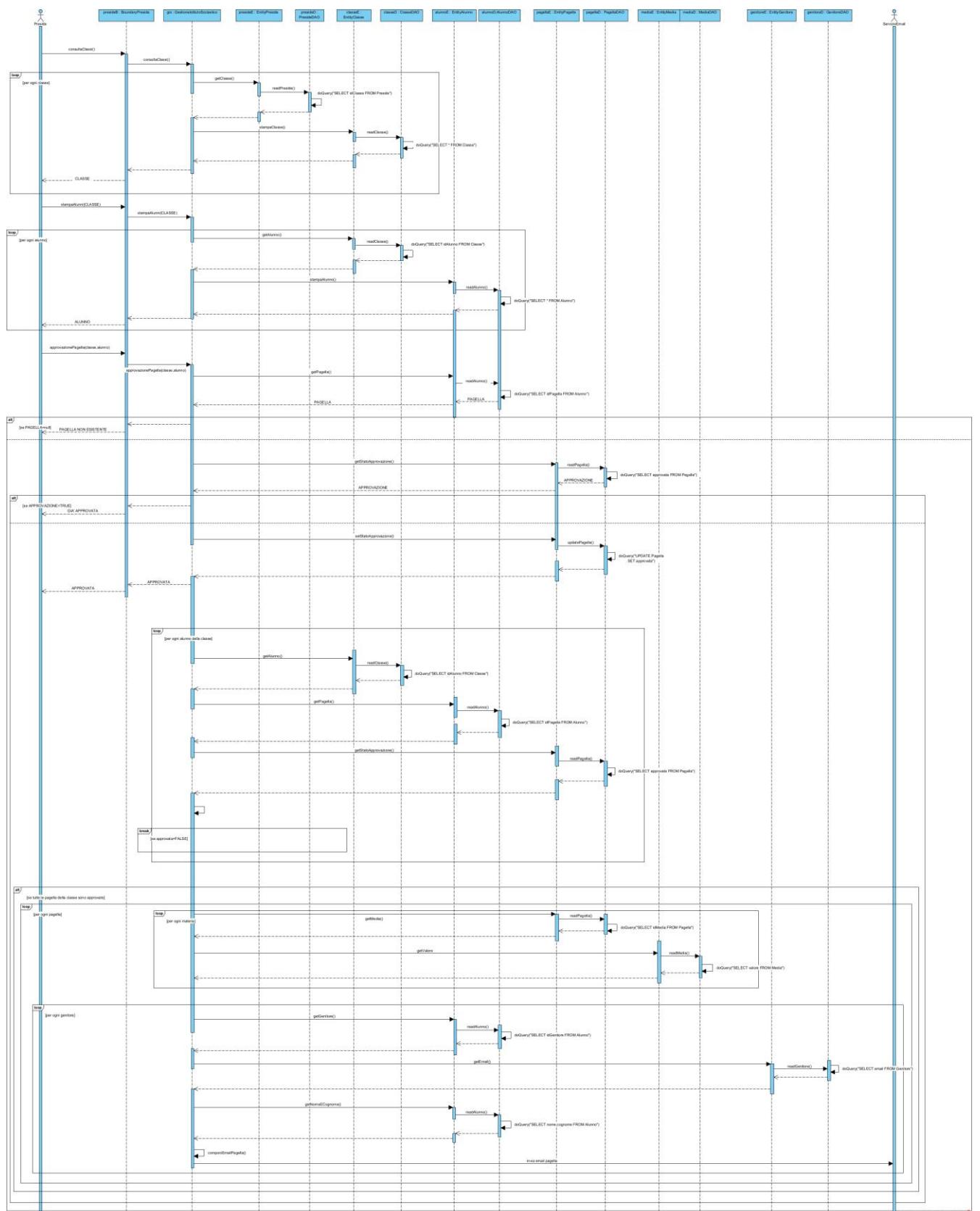
- Generazione Pagelle



- Modifica Pagella



• Approvazione Pagella



5. Implementazione

Come pattern architetturale è stato usato il BCED, un esempio di Layered Architecture. Il codice è diviso in 4 package, uno per ogni strato dell'architettura, chiamati “boundary”, “control”, “entity” e “database”.

1. Il package **“boundary”** contiene un JFrame chiamato “MainFrame” che deve essere eseguito per far partire l'applicazione e una serie di JDialog, “Alunno”, “Genitore”, “Docente”, “Segreteria”, “Preside”, “ConsultaRegistroDocente”, “InserisciAttivita”, “InserisciVoto”, “ModificaPagella”, che sono eventualmente istanziati dal “MainFrame”.
2. Il package **“control”** contiene una classe “GestioneIstitutoScolastico” che racchiude una serie di metodi che rappresentano la logica di business dell'applicazione. Queste funzioni sono “stampaClassiEAlunni”, “stampaClassiEMaterie”, “stampaAlunni”, “stampaFigli”, “inserimentoVoto”, “inserimentoAttivita”, “letturaAttivita”, “letturaCarriera”, “calcolaMedia”, “generazionePagelle”, “inserimentoUtente”, “modificaPagella” e “approvazionePagella”.
3. Il package **“entity”** contiene classi che rappresentano entità del dominio del problema, già definite in fase di analisi; le classi sono “EntityUtenteRegistrato”, “EntityUtente”, “EntityPreside”, “EntityAlunno”, “EntityGenitore”, “EntityDocente”, “EntityClasse”, “EntityRegistroElettronico”, “EntityAttivita”, “EntityValutazione”, “EntityVoto”, “EntityMedia”, “EntityInsegna” e “EntityPagella”.
4. Il package **“database”** è composto da classi che gestiscono l'accesso al database per ognuna delle entità; le classi sono “DBPreside”, “DBAlunno”, “DAGenitore”, “DBDocente”, “DBClasse”, “DBRegistroElettronico”, “DBAttivita”, “DBVoto”, “DBMedia”, “DBInsegna”, “DBPagella” e “DBConnectionManager”.

Oltre all'IDE Eclipse è stato necessario installare l'applicazione MySql per la realizzazione di un database per la persistenza dei dati. Per permettere ai due ambienti di comunicare, viene inserito nel Java project un connettore (mysql-connector-j-8.0.33) che si può osservare nelle Referenced Libraries. Inoltre, è stato utilizzato il software Window Builder per la costruzione delle interfacce grafiche.

In seguito, è riportata una porzione del javadoc prodotto per il class file di “GestioneIstitutoScolastico”.

Class GestioneIstitutoScolastico

[java.lang.Object](#)

control.GestioneIstitutoScolastico

public class GestioneIstitutoScolastico extends [Object](#)

- **Constructor Details**

- **GestioneIstitutoScolastico**

```
public GestioneIstitutoScolastico()
```

- **Method Details**

- **stampaClassiEAlunni**

public **String** stampaClassiEAlunni(**String** username)

Il metodo stampaClassiEAlunni permette di visualizzare l'anno e la sezione di tutte le classi dell'istituto scolastico e il nome, il cognome e l'username degli alunni che ne fanno parte. La funzione puo' essere eseguita solo dal preside.

Parameters:

username - del preside

Returns:

String che contiene le informazioni specificate

- **stampaClassiEMaterie**

public **String** stampaClassiEMaterie(**String** username)

Il metodo stampaClassiEMaterie permette di visualizzare l'anno e la sezione delle classi in cui insegna il docente e delle rispettive materie insegnate. La funzione e' accessibile solo ai docenti.

Parameters:

username - del docente

Returns:

String che contiene le informazioni desiderate

- **stampaAlunni**

public **String** stampaAlunni(**String** anno, **String** sezione)

Il metodo stampaAlunni permette di visualizzare il nome, il cognome e l'username degli alunni facenti parte di una classe (anno,sezione). La funzione e' accessibile solo ai docenti o al preside durante l'esecuzione di stampaClassiEAlunni.

Parameters:

anno - della classe

sezione - della casse

Returns:

String che contiene le informazioni specificate

- **stampaFigli**

public **String** stampaFigli(**String** username)

Il metodo stampaFigli permette di visualizzare il nome, il cognome e l'username degli alunni, figli di un genitore. La funzione puo' essere chiamata da un genitore.

Parameters:

username - del genitore

Returns:

String che contiene le informazioni richieste

o inserimentoVoto

```
public String inserimentoVoto(int valore, String materia, String data,  
String username)
```

Il metodo inserimentoVoto permette di inserire il voto (valore, media, data) di un alunno nel registro elettronico ed e' chiamato da un docente. Se il voto interito e' insufficiente, la funzione simula l'invio di una email ai genitori dell'alunno per notificarli.

Parameters:

valore - del voto
 materia - del voto
 data - del voto
 username - dell'alunno

Returns:

String che contiene un messaggio di errore o conferma

o inserimentoAttivita

```
public String inserimentoAttivita(String materia, String descrizione, String data,  
String anno, String sezione)
```

Il metodo inserimentoAttivita permette di inserire un'attivita' (materia, descrizione, data) svolta in una classe nel registro elettronico ed e' chiamato da un docente.

Parameters:

materia - dell'attivita'
 descrizione - dell'attivita'
 data - dell'attivita'
 anno - della classe
 sezione - della classe

Returns:

String che contiene un messaggio di errore o conferma

o letturaAttivita

```
public String letturaAttivita(String username, String data)
```

Il metodo letturaAttivita permette di visualizzare le attivita' (descrizione,materia) svolte in una certa data nella classe di un alunno. La funzione e' richiamata da un alunno.

Parameters:

username - dell'alunno
 data - delle attivita'

Returns:

String che contiene le informazioni desiderate

o letturaCarriera

```
public String letturaCarriera(String username)
```

Il metodo letturaCarriera permette di visualizzare i voti (valore,materia,data) e le medie (valore,materia) di un alunno, figlio di un genitore. La funzione e' eseguita da un genitore.

Parameters:

username - dell'alunno

Returns:

String che contiene le informazioni specificate

o **calcolaMedia**

private double calcolaMedia([ArrayList<Integer>](#) valori)

Il metodo calcolaMedia calcola la media di un insieme di interi. La funzione e' chiamata durante l'esecuzione dei metodi letturaCarriera e genezarionePagelle.

Parameters:

valori - di interi

Returns:

double che e' la media

o **generazionePagelle**

public void generazionePagelle()

Il metodo generazionePagelle genera per ogni alunno dell'istituto scolastico una pagella contenente le medie dei voti assegnati. La funzione e' chiamata allo scadere di un quadrimestre.

o **inserimentoUtente**

public [String](#) inserimentoUtente([String](#) username, [String](#) password,
[String](#) nome, [String](#) cognome, [String](#) residenza, [String](#) nascita,
[String](#) codiceFiscale, [Integer](#) cellulare, [String](#) email, [String](#) tipoUtente)

Il metodo inserimentoUtente permette di registrare un nuovo utente nel sistema ed e' svolto dalla segreteria. Bisogna specificare l'username, la password, il nome, il cognome, la residenza, la data di nascita, il codice fiscale, il cellulare, l'email e il tipo di utente. L'utente puo' essere un alunno, un docente o un genitore.

Parameters:

username - dell'utente

password - dell'utente

nome - dell'utente

cognome - dell'utente

residenza - dell'utente

nascita - dell'utente

codiceFiscale - dell'utente

cellulare - dell'utente

email - dell'utente

tipoUtente

Returns:

String che contiene un messaggio di errore o conferma

o **modificaPagella**

public [String](#) modificaPagella([String](#) username, [String](#) materia,
int arrotondamento)

Il metodo modificaPagella permette al docente di modificare il valore di una media (materia) della una pagella di un alunno. La modifica puo' consistere in un arrotondamento per eccesso (0) o per difetto (1). La funzione e' eseguibile solo da un docente.

Parameters:

username - dell'alunno
materia - della media
arrotondamento - per eccesso o difetto

Returns:

String che contiene un messaggio di errore o conferma

○ approvazionePagella

public [String](#) approvazionePagella([String](#) username)

Il metodo approvazionePagella permette di approvare la pagella di un alunno. Se tutte le pagelle degli altri alunni della classe risultano approvate, il metodo simula l'invio di una email ai genitori per notificarli. La funzione puo' essere chiamata solo dal preside.

Parameters:

username - dell'alunno

Returns:

String che contiene un messaggio di errore o conferma

File Java	LOC	~LLOC
GestionelIstitutoScolastico	398	169
DBAlunno	204	114
DBAttivita	112	44
DBClasse	97	47
DBConnectionManager	56	40
DBDocente	182	109
DBGenitore	180	104
DBInsegna	88	38
DBMedia	103	51
DBPagella	135	75
DBPreside	84	47
DBRegistroElettronico	138	99
DBVoto	114	79
EntityAlunno	116	81
EntityAttivita	93	64
EntityClasse	96	70
EntityDocente	81	61
EntityGenitore	81	59
EntityInsegna	59	33
EntityPagella	88	61
EntityMedia	72	51
EntityPreside	42	28
EntityRegistroElettronico	133	74

EntityUtente	97	42
EntityUtenteRegistrato	29	16
EntityValutazione	32	13
EntityVoto	69	36
Alunno	69	46
ConsultaRegistroDocente	83	57
Docente	83	52
Genitore	90	55
InserisciAttivita	115	81
InserisciVoto	94	74
MainFrame	120	71
ModificaPagella	84	68
Preside	88	74
Segreteria	179	136

	Stima delle LOC	LOC effettive
Registrazione Utente	1153	1211
Lettura Carriera	995	1251
Invio E-mail Pagella	901	1305

Per le funzionalità di “Lettura Carriera” e “Invio E-mail Pagella” la sottostima delle LOC necessarie per implementarle può essere dovuta al fatto che non è noto, a priori, il numero specifico di voti o di genitori di un alunno.

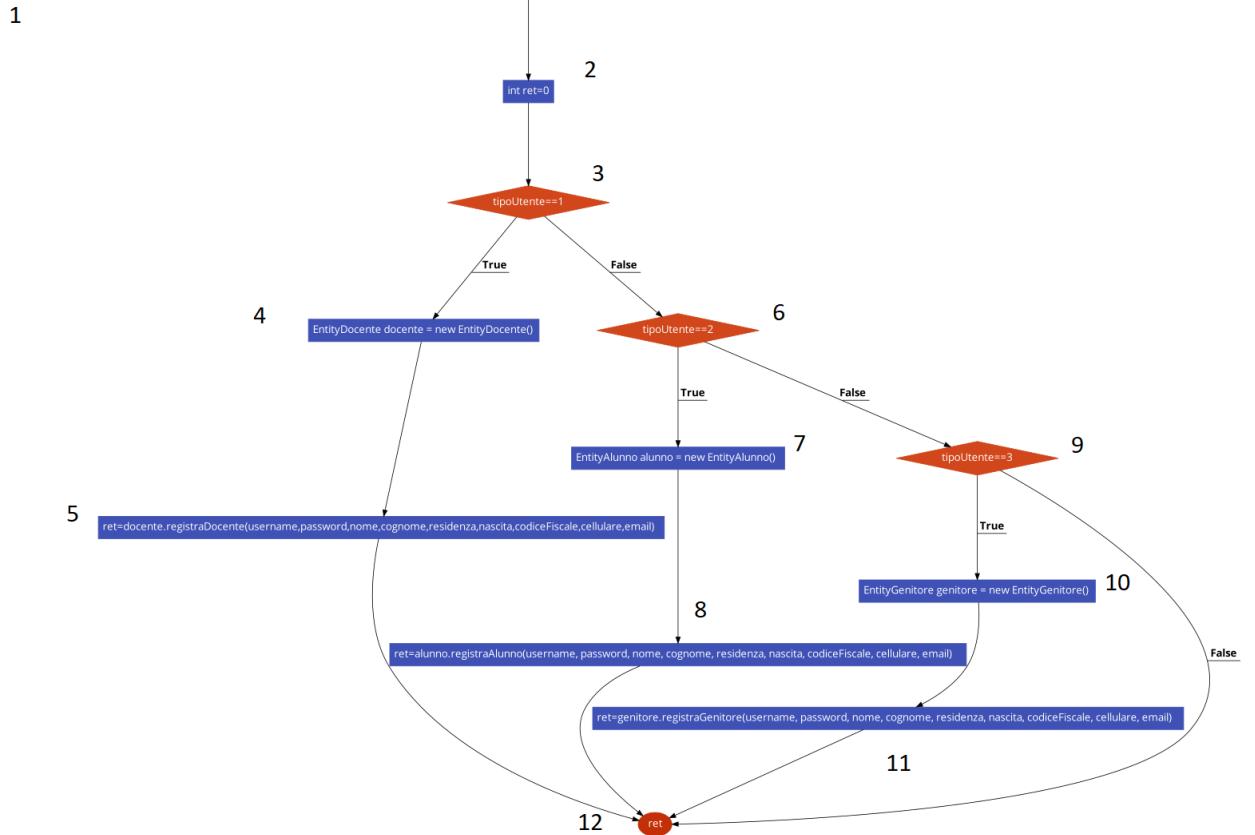
6. Testing

7.1 Test strutturale e Complessità ciclomatica

- REGISTRAZIONE UTENTE

```
public int inserimentoUtente(String username, String password, String nome, String cognome, String residenza, String nascita, String codiceFiscale, int cellulare, String email, int tipoUtente) {
    int ret=0;
    if(tipoUtente==1) {
        EntityDocente docente = new EntityDocente();
        ret=docente.registraDocente(username, password, nome, cognome, residenza, nascita, codiceFiscale, cellulare, email);
    }
    else if(tipoUtente==2) {
        EntityAlunno alunno = new EntityAlunno();
        ret=alunno.registraAlunno(username, password, nome, cognome, residenza, nascita, codiceFiscale, cellulare, email);
    }
    else if(tipoUtente==3) {
        EntityGenitore genitore = new EntityGenitore();
        ret=genitore.registraGenitore(username, password, nome, cognome, residenza, nascita, codiceFiscale, cellulare, email);
    }
    return ret;
}
```

public int inserimentoUtente(String username, String password, String nome, String cognome, String residenza, String nascita, String codiceFiscale, int cellulare, String email, int tipoUtente)



NUMERO CICLOMATICICO:

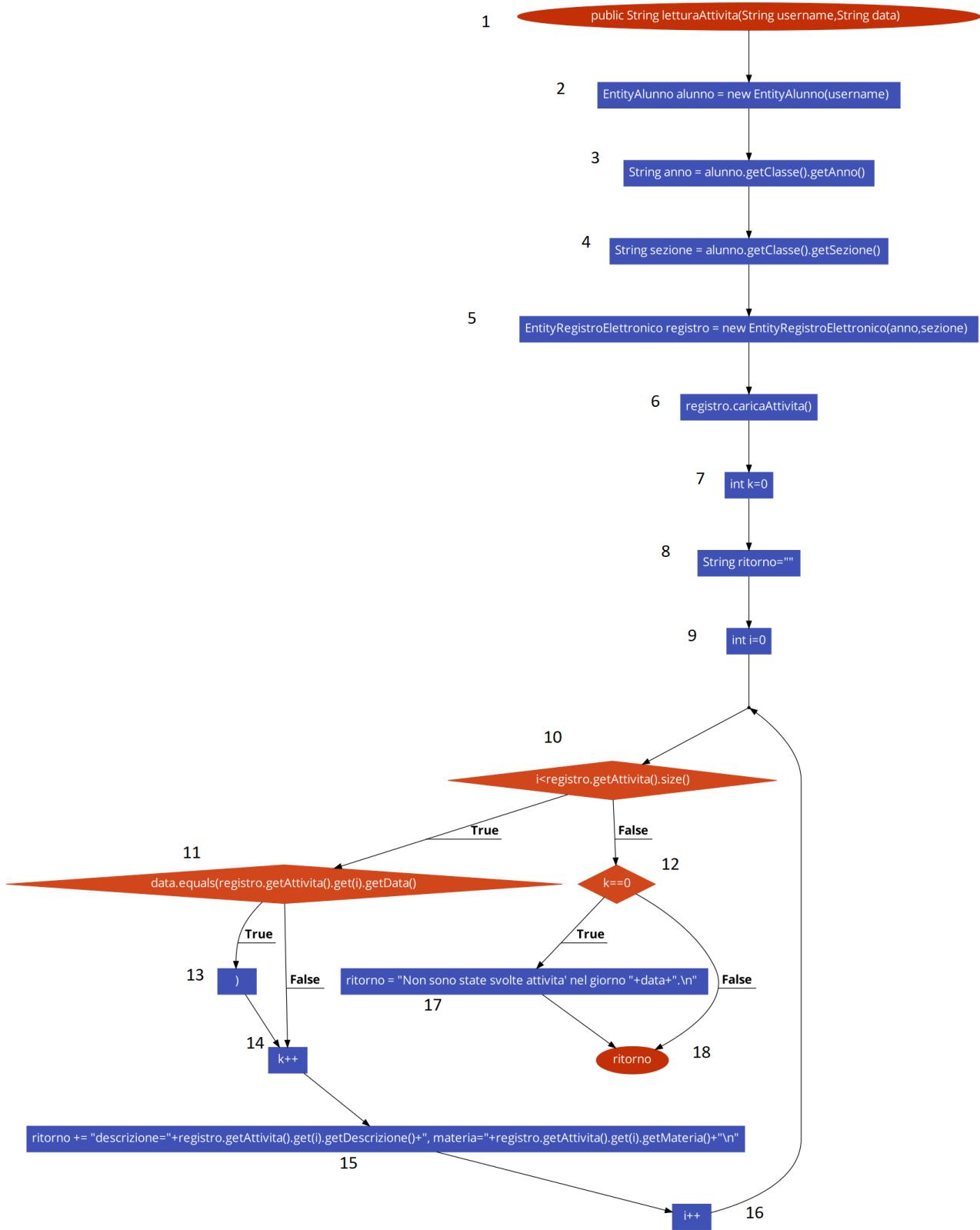
- numero di regioni chiuse del grafo + 1 = 3 + 1 = 4
- numero di nodi predicati + 1 = 3 + 1 = 4
- numero degli archi - numero dei nodi + 2 = 14 - 12 + 2 = 4

CAMMINI:

- 1) 1-2-3-4-5-12
- 2) 1-2-3-6-7-8-12
- 3) 1-2-3-6-9-10-11-12
- 4) 1-2-3-6-9-12

• LETTURA ATTIVITA

```
public String letturaAttivita(String username, String data) {  
    EntityAlunno alunno = new EntityAlunno(username);  
    String anno = alunno.getClasse().getAnno();  
    String sezione = alunno.getClasse().getSezione();  
    EntityRegistroElettronico registro = new  
EntityRegistroElettronico(anno, sezione);  
    registro.caricaAttivita();  
    int k=0;  
    String ritorno="";  
    for(int i=0;i<registro.getAttivita().size();i++) {  
        if(data.equals(registro.getAttivita().get(i).getData())) {  
            k++;  
            ritorno +=  
"descrizione="+registro.getAttivita().get(i).getDescrizione()+",  
materia="+registro.getAttivita().get(i).getMateria()+"\n";  
        }  
    }  
    if(k==0) ritorno = "Non sono state svolte attivita' nel giorno  
"+data+".\n";  
    return ritorno;  
}
```



NUMERO CICLOMATICO:

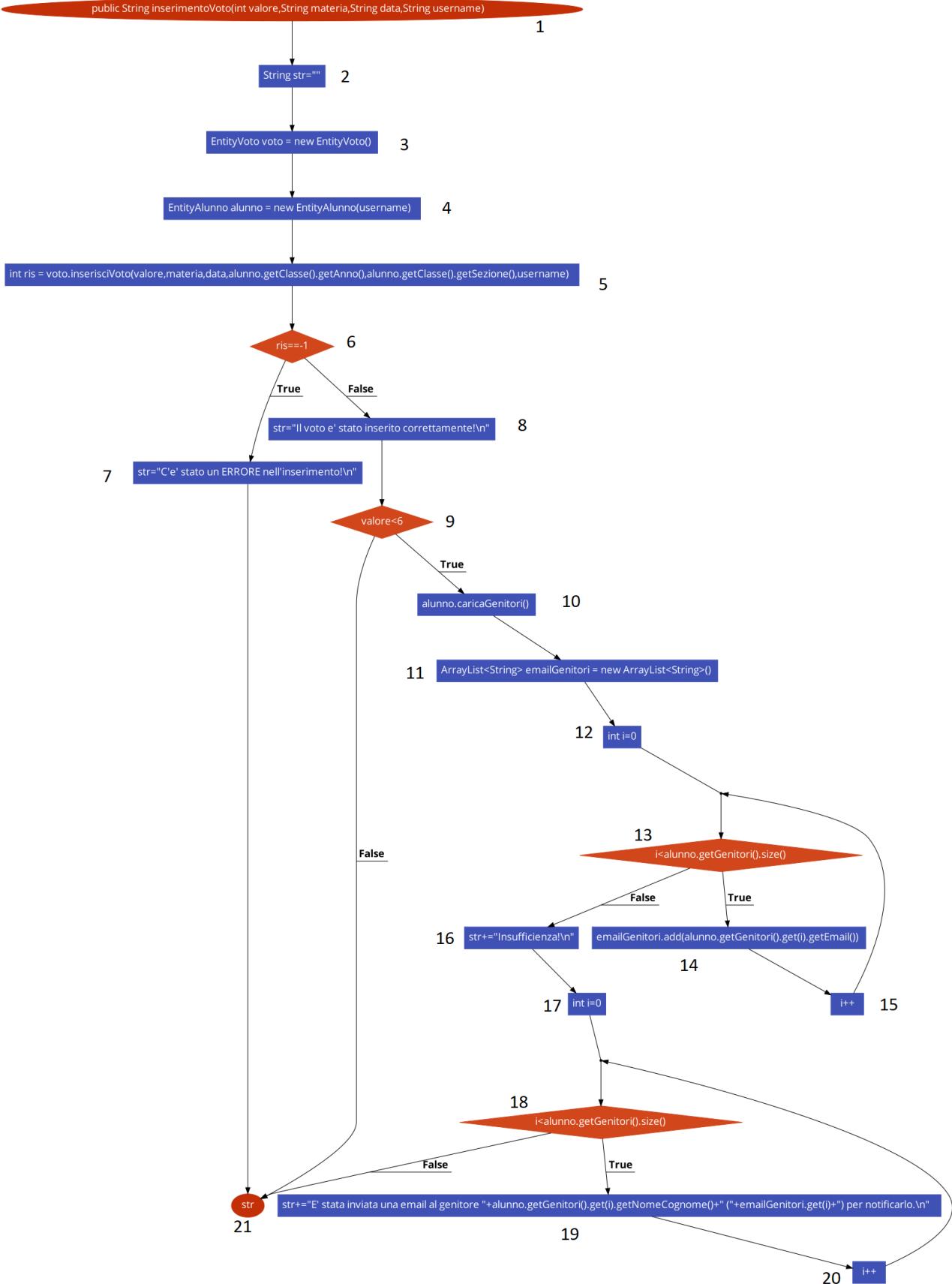
- numero di regioni chiuse del grafo + 1 = 3 + 1 = 4
- numero di nodi predicati + 1 = 3 + 1 = 4
- numero degli archi – numero dei nodi + 2 = 19 - 17 + 2 = 4

CAMMINI:

- 1) 1-2-3-4-5-6-7-8-9-10-11-13-14-15-16-10-12-17-18
- 2) 1-2-3-4-5-6-7-8-9-10-11-14-15-16-10-12-17-18
- 3) 1-2-3-4-5-6-7-8-9-10-11-14-15-16-10-12-18
- 4) 1-2-3-4-5-6-7-8-9-10-12-18

- INSERIMENTO VOTO

```
public String inserimentoVoto(int valore, String materia, String data, String username) {
    String str="";
    EntityVoto voto = new EntityVoto();
    EntityAlunno alunno = new EntityAlunno(username);
    int ris =
voto.inserisciVoto(valore, materia, data, alunno.getClasse().getAnno(), alunno.getClasse().getSezione(), username);
    if(ris== -1) str="C'e' stato un ERRORE nell'inserimento!\n";
    else {
        str="Il voto e' stato inserito correttamente!\n";
        if(valore<6) {
            alunno.caricaGenitori();
            ArrayList<String> emailGenitori = new
ArrayList<String>();
            for(int i=0;i<alunno.getGenitori().size();i++) {
                emailGenitori.add(alunno.getGenitori().get(i).getEmail());
            }
            str+="Insufficienza!\n";
            for(int i=0;i<alunno.getGenitori().size();i++) {
                str+="E' stata inviata una email al genitore
"+alunno.getGenitori().get(i).getNomeCognome()+" ("+emailGenitori.get(i)+") per
notificarlo.\n";
            }
        }
    }
    return str;
}
```



NUMERO CICLOMATICICO:

- numero di regioni chiuse del grafo + 1 = 4 + 1 = 5
- numero di nodi predicati + 1 = 4 + 1 = 5
- numero degli archi - numero dei nodi + 2 = 24 - 21 +2 = 5

CAMMINI:

- 1) 1-2-3-4-5-6-7-21
- 2) 1-2-3-4-5-6-7-8-9-21
- 3) 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-13-16-17-18-21
- 4) 1-2-3-4-5-6-7-8-9-10-11-12-13-16-17-18-21
- 5) 1-2-3-4-5-6-7-8-9-10-11-12-13-16-17-18-19-20-18-21

7.2 Test funzionale

In seguito, è riportata una parte dei risultati dell'esecuzione dei test funzionali precedentemente pianificati e le eventuali azioni di *debugging* svolte a seguito di *FAIL*. I test case non menzionati sono simili a quelli considerati.

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	Nome valido Cognome valido Nascita valida Codice Fiscale valido Residenza valida Email valida Cellulare valido Username valida Password valida	L'utente non è stato ancora registrato.	{Nome: "Mario", Cognome: "Rossi", Data: 10/11/2002, CodiceFiscale: "RSSNMO02S10F839S", Residenza: "Napoli", Email: " mariorossi@gmail.com ", Cellulare: 345678912, Username: "mariorossi", Password: "mariocalcetto12"}	L'utente e' stato inserito correttamente.	È stato inserito un nuovo utente nel sistema.	L'utente e' stato inserito correttamente.	L'utente e' stato inserito correttamente.	PASS
7	Codice Fiscale- stringa di caratteri di lunghezza diversa da 16	Nome valido Cognome valido Nascita valida Codice Fiscale [ERROR] Residenza valida		{Nome: "Mario", Cognome: "Rossi", Data: 10/11/2002, CodiceFiscale: "RSSNMO02S10F83555559SAS16T", Residenza: "Napoli", Email: "mariorossi@gmail.com", Cellulare: 345678912, Username: "mariorossi", Password: "mariocalcetto12"}	Il codice fiscale non e' composto da 16 caratteri. ERRORE!	Non è stato aggiunto nessun utente nel sistema.	L'utente e' stato inserito correttamente.	L'utente e' stato inserito un nuovo utente nel sistema.	FAIL

		Email valida Cellulare valido Username valida Password valida							
Viene inserito un controllo di validità sulla lunghezza del codice fiscale nel metodo inserimentoUtente della classe control.GestionelIstitutoScolastico.									
7	Codice Fiscale- stringa di caratteri di lunghezza diversa da 16	Nome valido Cognome valido Nascita valida Codice Fiscale [ERROR] Residenza valida Email valida Cellulare valido Username valida Password valida		{Nome: "Mario", Cognome: "Rossi", Data: 10/11/2002, CodiceFiscale: "RSSNMO02S10F83555559SAS16T", Residenza: "Napoli", Email: "mariorossi@gmail.com", Cellulare: 345678912, Username: "mariorossi", Password: "mariocalcetto12"}	Il codice fiscale non e' composto da 16 caratteri. ERRORE!	Non è stato aggiunto nessun utente nel sistema	Il codice fiscale non e' composto da 16 caratteri. ERRORE!	Il codice fiscale non e' composto da 16 caratteri. ERRORE!	PASS
10	Email - stringa in cui non è presente il simbolo @	Nome valido Cognome valido		{Nome: "Mario", Cognome: "Rossi", Data: 10/11/2002, CodiceFiscale: "RSSNMO02S10F839S",	L'email non contiene la chiocciola. ERRORE!	Non è stato aggiunto nessun	L'utente e' stato inserito correttamente.	È stato inserito un nuovo utente nel sistema.	FAIL

		Nascita valida Codice Fiscale valido Residenza valida Email [ERROR] Cellulare valido Username valida Password valida		Residenza: "Napoli", Email: "marioross@gmail.com", Cellulare: 345678912, Username: "mariorossi", Password: "mariocalcetto12"}		utente nel sistema		
--	--	--	--	---	--	-----------------------	--	--

Viene inserito un controllo di validità sul formato dell'e-mail nel metodo inserimentoUtente della classe control.GestionelIstitutoScolastico.

10	Email - stringa in cui non è presente il simbolo @	Nome valido Cognome valido Nascita valida Codice Fiscale valido Residenza valida Email [ERROR] Cellulare valido		{Nome: "Mario", Cognome: "Rossi", Data: 10/11/2002, CodiceFiscale: "RSSNMO02S10F839S", Residenza: "Napoli", Email: "marioross@gmail.com", Cellulare: 345678912, Username: "mariorossi", Password: "mariocalcetto12"}	L'email non contiene la chiocciola. ERRORE!	Non è stato aggiunto nessun utente nel sistema	L'email non contiene la chiocciola. ERRORE!	Non è stato aggiunto nessun utente nel sistema	PASS
----	--	---	--	---	--	---	--	---	------

		Username valida Password valida							
3	Nome – stringa che contiene simboli che non sono caratteri	Nome [ERROR] Cognome valido Nascita valido Codice Fiscale valido Residenza valida Email valida Cellulare valido Username valida Password valida		{Nome: "(!?)mario@ç", Cognome: "Rossi", Data: 10/11/2002, CodiceFiscale: "RSSNMO02S10F839S", Residenza: "Napoli", Email: "mariorossi@gmail.com", Cellulare: 345678912, Username: "mariorossi", Password: "mariocalcetto12"}	Il nome contiene numeri o simboli. ERRORE!	Non è stato aggiunto nessun utente nel sistema	L'utente e' stato inserito correttamente.	È stato inserito un nuovo utente nel sistema.	FAIL

Viene inserito un controllo di validità sul formato del nome nel metodo inserimentoUtente della classe control.GestionelIstitutoScolastico.

3	Nome – stringa che contiene simboli che non sono caratteri	Nome [ERROR] Cognome valido Nascita valido		{Nome: "(!?)mario@ç", Cognome: "Rossi", Data: 10/11/2002, CodiceFiscale: "RSSNMO02S10F839S", Residenza: "Napoli", Email: "mariorossi@gmail.com", Cellulare: 345678912,	Il nome contiene numeri o simboli. ERRORE!	Non è stato aggiunto nessun utente nel sistema	Il nome contiene numeri o simboli. ERRORE!	Non è stato aggiunto nessun utente nel sistema	PASS
---	--	--	--	---	---	--	---	--	------

		Codice Fiscale valido Residenza valida Email valida Cellulare valido Username valida Password valida		Username: "mariorossi", Password: "mariocalcetto12"} 				
6	Nascita - formato non valido	Nome valido Cognome valido Nascita [ERROR] Codice Fiscale valido Residenza valida Email valida Cellulare valido Username valida Password valida		{Nome: "Mario", Cognome: "Rossi", Data: dieci/13/2002, CodiceFiscale: "RSSNMO02S10F839S", Residenza: "Napoli", Email: "mariorossi@gmail.com", Cellulare: 345678912, Username: "mariorossi", Password: "mariocalcetto12"}	Il formato della data di nascita non e' valido. ERRORE!	Non è stato aggiunto nessun utente nel sistema	L'utente e' stato inserito correttamente.	È stato inserito un nuovo utente nel sistema.
Viene inserito un controllo di validità sul formato della data di nascita nel metodo inserimentoUtente della classe <u>control.GestionelIstitutoScolastico</u> .								

		Username valido Password [ERROR]						
--	--	---	--	--	--	--	--	--

Viene inserito un controllo di validità sulla lunghezza della password nel metodo inserimentoUtente della classe control.GestionelstitutoScolastico.

14	Password - stringa di caratteri di lunghezza > 30	Nome valido Cognome valido Nascita valida Codice Fiscale valido Residenza valida Email valida Cellulare valido Username valido Password [ERROR]		{Nome: "Mario", Cognome: "Rossi", Data: 10/11/2002, CodiceFiscale: "RSSNMO02S10F839S", Residenza: "Napoli", Email: "mariorossi@gmail.com", Cellulare: 345678912, Username: "mariorossi", Password: "mariocalcetto12SSSS SSSSSSSSSSSSSSSSSSSSSSSSSSSS SSSSSSSSSSSSSSSSSSSSSSSSSSSS SSSSSSSSSSSSSSSSSSSSSSSSSS"} 	La password e' troppo lunga. ERRORE!	Non è stato aggiunto nessun utente nel sistema	La password e' troppo lunga. ERRORE!	Non è stato aggiunto nessun utente nel sistema	PASS
11	Cellulare - numero intero < 1000000000 [ERROR]	Nome valido Cognome valido Nascita valida		{Nome: "Mario", Cognome: "Rossi", Data: 10/11/2002, CodiceFiscale: "RSSNMO02S10F839S", Residenza: "Napoli", Email: "mariorossi@gmail.com", Cellulare: 342,	Il cellulare e' troppo corto. ERRORE!	Non è stato aggiunto nessun utente nel sistema	L'utente e' stato inserito correttamente.	È stato inserito un nuovo utente nel sistema.	FAIL

		Codice Fiscale valido Residenza valida Email valida Cellulare [ERROR] Username valida Password valida		Username: "mariorossi", Password: "mariocalcetto12"} Viene inserito un controllo di validità sulla lunghezza del cellulare nel metodo inserimentoUtente della classe control.GestionelstitutoScolastico.					
11	Cellulare - numero intero < 1000000000 [ERROR]	Nome valido Cognome valido Nascita valida Codice Fiscale valido Residenza valida Email valida Cellulare [ERROR] Username valida Password valida		{Nome: "Mario", Cognome: "Rossi", Data: 10/11/2002, CodiceFiscale: "RSSNMO02S10F839S", Residenza: "Napoli", Email: "mariorossi@gmail.com", Cellulare: 342, Username: "mariorossi", Password: "mariocalcetto12"}	Il cellulare e' troppo corto. ERRORE!	Non è stato aggiunto nessun utente nel sistema	Il cellulare e' troppo corto. ERRORE!	Non è stato aggiunto nessun utente nel sistema	PASS

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	Data valido Valore valido		{Data: "20/05/2022", Valore: 7}	Il voto e' stato inserito correttamente!	È stato inserito un nuovo voto nel sistema.	Il voto e' stato inserito correttamente!	È stato inserito un nuovo voto nel sistema.	PASS
4	Valore – numero intero > 10	Data valido Valore [ERROR]		{Data: "20/05/2022", Valore: 34}	Il valore del voto non e' valido!	Non è stato aggiunto nessun voto nel sistema.	Il voto e' stato inserito correttamente!	È stato inserito un nuovo voto nel sistema.	FAIL
Viene inserito un controllo di validità sul valore del voto in boundary. InserisciVoto.									
4	Valore – numero intero >10	Data valido Valore [ERROR]		{Data: "20/05/2022", Valore: 34}	Il valore del voto non e' valido!	Non è stato aggiunto nessun voto nel sistema.	Il valore del voto non e' valido!	Non è stato aggiunto nessun voto nel sistema.	PASS
3	Valore – numero intero < 0	Data valido Valore [ERROR]		{Data: "20/05/2022", Valore: -3}	Il valore del voto non e' valido!	Non è stato aggiunto nessun voto nel sistema.	Il valore del voto non e' valido!	Non è stato aggiunto nessun voto nel sistema.	PASS

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	Data valida Descrizione valida		{Data: "20/05/2022", Descrizione: "Lettura Dante"}	L'attivita' e' stata inserita correttamente!	È stato inserito una nuova attività nel sistema.	L'attivita' e' stata inserita correttamente!	È stato inserito una nuova attività nel sistema.	PASS

Gli errori si sarebbero potuti osservare anche attraverso testing strutturale impiegando il framework JUnit. Ad esempio, per il test case n.ro 3 della test suite “Inserimento Utente” si sarebbe potuto progettare ed eseguire il seguente codice.

```

@Test
void testInserimentoUtente () {
    GestioneIstitutoScolastico g = new GestioneIstitutoScolastico ();
    String
ris=g.inserimentoUtente("alunno39","passw","Mario@39","Rossi","Napoli","12/02/2002","ASDQWE123456GFDE",1234567890,"m
arioross@gmail.com","alunno");
    assertEquals("L'utente e' stato inserito correttamente\n",ris);
}

```