



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



**CONSERVATORY**  
MANAGEMENT SOFTWARE

***SISTEMA SOFTWARE PER LA GESTIONE  
DI UN CONSERVATORIO***

<b>1. Specifiche informali</b>	<b>4</b>
<b>2. Analisi e specifica dei requisiti</b>	<b>5</b>
2.1 Analisi nomi-verbi	5
2.2 Revisione dei Requisiti	6
2.3 Glossario dei termini	7
2.4 Classificazione dei requisiti	8
2.4.1 Requisiti funzionali:	8
2.4.2 Requisiti sui dati	9
2.4.3 Vincoli/Altri requisiti	9
2.5 Modellazione dei casi d'uso	10
2.5.1 Attori e casi d'uso	10
2.5.2 Diagramma dei casi d'uso	13
2.5.3 Scenari	14
2.6 Diagramma delle classi	18
2.6.1 Responsabilità	19
2.7 Sequence Diagrams	20
<b>3. Piano di test funzionale</b>	<b>24</b>
3.1 Autenticazione	24
3.2 InserisciVerbale	26
3.3 Aggiungi Docente	28
3.4 CreaPianoDiStudio	30
<b>4. Progettazione</b>	<b>32</b>
4.1: Diagramma delle classi (modello BCED)	32
4.1.2: Pattern BCED	33
4.1.2.1 Package Boundary	33
4.1.2.2 Package Controller	33
4.1.2.3 Package Entity	34
4.1.2.4 Package Database	34
4.2 Diagrammi di sequenza	36
4.2.1 Autenticazione	36
4.2.2 CreaPianoDiStudi	37
4.2.3 VerbalizzaEsame	37
4.2.4 AggiungiDocente	38
<b>5. Implementazione</b>	<b>39</b>
5.1 Package Database	39

5.2 Package Entity	39
5.3 Package Controller	40
5.4 Package Boundary	40
5.5 Package DTO	41
5.6 Diagramma di deployment	41
<b>6. Testing</b>	<b>42</b>
6.1 Test strutturale	42
6.1.1 Autenticazione	43
6.1.2 AggiungiDocente	44
6.1.3 SalvaPianoDiStudi	46
6.1.4 CreaVerbale	47

# 1. Specifiche informali

Il sistema prevede cinque tipologie di corsi di studio: canto, strumento, composizione, musica d'insieme e didattica della musica. Per i corsi di canto, sono offerti i corsi di studio in canto lirico e canto jazz. Per i corsi di studi in strumento sono disponibili i corsi di pianoforte, violino, chitarra, ecc. Un corso può essere propedeutico a zero o più corsi.

Gli studenti possono scegliere di aderire a piani di studio personalizzati, scegliendo un insieme di corsi da includere nel proprio piano di studi. A tal fine, lo studente deve autenticarsi al sistema, visualizzare l'insieme dei corsi disponibili, e creare il proprio piano di studi personalizzato aggiungendo ad esso un insieme di corsi a scelta.

La segreteria studenti utilizza il sistema per aggiungere e modificare docenti e corsi, e in ogni anno accademico per associare un docente all'edizione del corso di quell'anno. Un corso ha un codice, una denominazione e un numero di crediti formativi (CFU). I docenti hanno una matricola, un cognome e un nome. (Per semplicità, non è prevista la suddivisione di un corso in più cattedre.)

I dati degli studenti iscritti (comprese le loro credenziali di accesso, username, password e PIN) provengono dall'anagrafe degli studenti del Conservatorio: periodicamente, il sistema preleva i dati dal sistema Anagrafe Studenti.

Il sistema permette ai docenti di mantenere un registro che memorizza gli esami sostenuti dagli studenti per un determinato corso, riportando la valutazione in trentesimi, l'eventuale lode, le note del docente in testo libero e la data del superamento dell'esame.

Una volta completato un appello di esami in una determinata data, il docente procede alla verbalizzazione degli esami superati aprendo una "camicia elettronica" (lista degli studenti esaminati con voto) che costituisce il verbale e riporta un codice identificativo univoco, la data dell'appello, ed il codice del corso. L'inserimento degli esami superati nel verbale è effettuato dai docenti, i quali inseriranno la matricola ed il voto per ciascuno studente.

All'atto della chiusura della camicia, il sistema verifica per ognuno degli studenti il conseguimento da parte dello studente degli esami propedeutici. Nel caso in cui almeno un esame propedeutico non sia stato sostenuto da uno studente, il sistema annulla l'esame e fornisce un messaggio di errore, altrimenti gli esami vengono memorizzati nel sistema.

Gli studenti possono accedere al sistema per visualizzare gli esami sostenuti con i voti. Inoltre, possono visualizzare la media aritmetica e la media ponderata.

## 2. Analisi e specifica dei requisiti

### 2.1 Analisi nomi-verbi

Il sistema prevede cinque tipologie di **corsi di studio**: canto, strumento, composizione, musica d'insieme e didattica della musica. Per i corsi di canto, sono offerti i corsi di studio in canto lirico e canto jazz. Per i corsi di studi in strumento sono disponibili i corsi di pianoforte, violino, chitarra, ecc. Un corso può essere propedeutico a zero o più corsi.

Gli **studenti** possono scegliere di aderire a **piani di studio** personalizzati, scegliendo un insieme di corsi da includere nel proprio piano di studi. A tal fine, lo studente deve autenticarsi al sistema, visualizzare l'insieme dei corsi disponibili, e creare il proprio piano di studi personalizzato aggiungendo ad esso un insieme di corsi a scelta.

La **segreteria studenti** utilizza il sistema per aggiungere e modificare docenti e corsi, e in ogni anno accademico per associare un docente all'edizione del corso di quell'anno. Un corso ha un **codice**, una **denominazione** e un numero di crediti formativi (**CFU**). I docenti hanno una **matricola**, un **cognome** e un **nome**. (Per semplicità, non è prevista la suddivisione di un corso in più cattedre.)

I dati degli studenti iscritti (comprese le loro credenziali di accesso, **username**, **password** e **PIN**) provengono dall'anagrafe degli studenti del Conservatorio: periodicamente, il sistema preleva i dati dal sistema Anagrafe Studenti.

Il sistema permette ai docenti di mantenere un **registro** che memorizza gli **esami sostenuti** dagli studenti per un determinato corso, riportando la **valutazione in trentesimi**, **l'eventuale lode**, le **note del docente** in testo libero e la **data del superamento dell'esame**.

Una volta completato un appello di esami in una determinata data, il **docente** procede alla verbalizzazione degli esami superati aprendo una "**camicia elettronica**" (lista degli studenti esaminati con voto) che costituisce il verbale e riporta un **codice identificativo univoco**, la **data dell'appello**, ed il **codice del corso**. L'inserimento degli esami superati nel verbale è effettuato dai docenti, i quali inseriranno la matricola ed il voto per ciascuno studente.

All'atto della chiusura della camicia, il sistema verifica per ognuno degli studenti il conseguimento da parte dello studente degli esami propedeutici. Nel caso in cui almeno un esame propedeutico non sia stato sostenuto da uno studente, il sistema annulla l'esame e fornisce un messaggio di errore, altrimenti gli esami vengono memorizzati nel sistema.

Gli studenti possono accedere al sistema per visualizzare gli esami sostenuti con i voti. Inoltre, possono visualizzare la media aritmetica e la media ponderata.

LEGENDA:

**Classe**

**Attributo**

**Funzionalità**

**Attore**

**Classe-Attore**

## 2.2 Revisione dei Requisiti

1. Il Sistema deve offrire allo studente una funzionalità per autenticarsi.
2. Il Sistema deve offrire allo studente una funzionalità per visualizzare l'insieme dei corsi disponibili.
3. Il Sistema deve offrire allo studente una funzionalità per creare il proprio piano di studi personalizzato.
4. Il Sistema deve offrire allo studente una funzionalità per visualizzare gli esami sostenuti con voto.
5. Il Sistema deve offrire allo studente una funzionalità per visualizzare la propria media.
6. Il Sistema deve offrire allo studente una funzionalità per visualizzare la propria media ponderata.
7. Il Sistema deve offrire alla segreteria studenti una funzionalità per aggiungere docenti e corsi.
8. Il Sistema deve offrire alla segreteria studenti una funzionalità per modificare docenti e corsi.
9. Il Sistema deve offrire alla segreteria studenti una funzionalità per associare un docente all'edizione del corso di quell'anno.
10. Di ogni Corso di Studio si vuole memorizzare codice, denominazione numero di crediti formativi (CFU).
11. Di ogni Docente si vuole memorizzare matricola, cognome e nome.
12. Di ogni Studente iscritto si vuole memorizzare username, password e PIN.
13. Il sistema prevede che periodicamente vengano prelevati i dati dal sistema Anagrafe Studenti.
14. Il Sistema deve offrire al docente una funzionalità per aggiornare un registro dei voti.
15. Di ogni Esame sostenuto si vuole memorizzare la valutazione in trentesimi, l'eventuale lode, le note del docente e la data del suo superamento.
16. Il Sistema deve offrire al docente una funzionalità per creare una verbale.
17. Di ogni verbale si vuole memorizzare un codice identificativo univoco, la data dell'appello, ed il codice del corso.
18. Il Sistema deve offrire al docente una funzionalità per inserire gli esami superati nel verbale, riportando matricola e voto per ciascuno studente.
19. Il Sistema deve verificare per ognuno degli studenti il conseguimento degli esami propedeutici.
20. Il Sistema deve annullare l'esame e fornire un messaggio di errore nel caso in cui almeno un esame propedeutico non sia stato sostenuto da uno studente.
21. Il Sistema deve memorizzare l'esame nel caso in cui tutti gli esami propedeutici siano stati sostenuti da uno studente.

## 2.3 Glossario dei termini

---

Termine	Descrizione (significato)	Sinonimi
Corso di studio	Tipologia di formazione offerta (canto, strumento, composizione, musica d'insieme, didattica della musica)	Percorso formativo
Corso	Insegnamento specifico con codice, denominazione e numero di CFU	Insegnamento
Canto lirico / Canto jazz	Sottotipologie dei corsi di canto	-
CFU (Crediti Formativi Universitari)	Unità di misura del carico di lavoro richiesto per il corso	Crediti
Piano di studi personalizzato	Insieme di corsi selezionati dallo studente per comporre il proprio piano formativo	Piano di studi
Docente	Persona incaricata dell'insegnamento di un corso; ha una matricola, un nome e un cognome	Professore
Edizione del corso	Offerta specifica del corso in un determinato anno accademico	Anno del corso
Matricola docente	Codice identificativo univoco assegnato al docente	ID docente
Matricola studente	Codice identificativo univoco assegnato allo studente	ID studente
Registro esami	Registro mantenuto dal docente con esiti degli esami sostenuti dagli studenti	Registro voti
Verbalizzazione	Procedura di registrazione ufficiale degli esami superati da parte del docente	Chiusura esami

Camicia elettronica	Documento digitale contenente la lista degli studenti esaminati con voti, associata a un codice, una data e un corso	Verbale d'esame
Propedeuticità	Condizione per cui un corso o esame deve essere superato prima di accedere ad altri corsi/esami	Requisito preliminare
Anagrafe Studenti	Sistema esterno da cui provengono i dati degli studenti iscritti	Database studenti
Autenticazione	Processo mediante cui lo studente accede al sistema inserendo credenziali (username, password, PIN)	Login
Media aritmetica	Media dei voti ottenuti dagli studenti calcolata sommando i voti e dividendo per il numero degli esami	Media semplice
Media ponderata	Media dei voti pesati per i CFU di ciascun esame	Media pesata
Sistema	Applicazione informatica usata da studenti, docenti e segreteria per la gestione dei corsi, piani di studio ed esami	Portale, piattaforma

---

## 2.4 Classificazione dei requisiti

### 2.4.1 Requisiti funzionali:

**RF01:** Il sistema deve offrire allo studente una funzionalità per autenticarsi.

**RF02:** Il sistema deve offrire allo studente una funzionalità per visualizzare l'insieme dei corsi disponibili.

**RF03:** Il sistema deve offrire allo studente una funzionalità per creare il proprio piano di studi personalizzato.

**RF04:** Il sistema deve offrire allo studente una funzionalità per visualizzare gli esami sostenuti con i voti.

**RF05:** Il sistema deve offrire allo studente una funzionalità per visualizzare la media aritmetica.

**RF06:** Il sistema deve offrire allo studente una funzionalità per visualizzare la media ponderata.

**RF07:** Il sistema deve offrire alla segreteria studenti una funzionalità per aggiungere i docenti e corsi.

**RF08:** Il sistema deve offrire alla segreteria studenti una funzionalità per modificare i docenti e corsi.

**RF09:** Il sistema deve offrire alla segreteria studenti una funzionalità per associare un docente all'edizione del corso di quell'anno.

**RF10:** Il sistema prevede che periodicamente vengano prelevati i dati dal sistema Anagrafe Studenti.

**RF11:** Il sistema deve offrire al docente una funzionalità per inserire un esame nel proprio registro dei voti.



**RF12:** Il Sistema deve offrire al docente una funzionalità per creare un verbale.

**RF13:** Il Sistema deve offrire al docente una funzionalità per inserire gli esami superati nel verbale, riportando matricola e voto per ciascuno studente.

**RF14:** Il Sistema deve offrire una funzionalità per verificare che gli esami inseriti all'interno del verbale siano validi rispetto alle propedeuticità del corso.

**RF15:** Il Sistema deve offrire al docente una funzionalità per memorizzare un esame validato.

**RF16:** Il Sistema deve offrire una funzionalità per chiudere un verbale salvandolo in memoria.

## 2.4.2 Requisiti sui dati

**RD01:** Di ogni Corso di Studio si vuole memorizzare codice, denominazione, numero di crediti formativi (CFU).

**RD02:** Di ogni Docente si vuole memorizzare matricola, cognome e nome.

**RD03:** Di ogni Studente iscritto si vuole memorizzare username, password e PIN.

**RD04:** Di ogni Esame sostenuto si vuole memorizzare la valutazione in trentesimi, l'eventuale lode, le note del docente e la data del suo superamento.

**RD05:** Di ogni camicia elettronica si vuole memorizzare un codice identificativo univoco, la data dell'appello, ed il codice del corso.

## 2.4.3 Vincoli/Altri requisiti

**RNF01:** Il sistema da sviluppare deve essere una applicazione web, accessibile da Chrome e Firefox su dispositivi desktop e tablet.

**RNF02:** Il sistema deve supportare almeno 100 accessi simultanei, con picchi concentrati nei periodi di iscrizione e verbalizzazione esami.

**RNF03:** La base utenti prevista è di circa 1.500 studenti e 80 docenti; ogni studente effettua in media 10 accessi a settimana, con una frequenza stimata di 300 interrogazioni/ora e 20 modifiche/giorno.

**RNF04:** Le operazioni di consultazione (es. visualizzazione corsi o voti) devono avere un tempo di risposta inferiore a 2 secondi nel 95% dei casi.

**RNF05:** L'accesso al sistema deve avvenire tramite autenticazione sicura (username, password, PIN), e tutte le comunicazioni devono avvenire su connessione HTTPS.

**RNF06:** Il sistema deve integrarsi con il sistema esterno *Anagrafe Studenti*, importando automaticamente i dati degli studenti una volta a settimana.

## 2.5 Modellazione dei casi d'uso

### 2.5.1 Attori e casi d'uso

#### Attori Primari

- Studente
- Docente
- Segreteria Studenti
- Tempo

#### Attori Secondari:

- Sistema Anagrafe Studenti

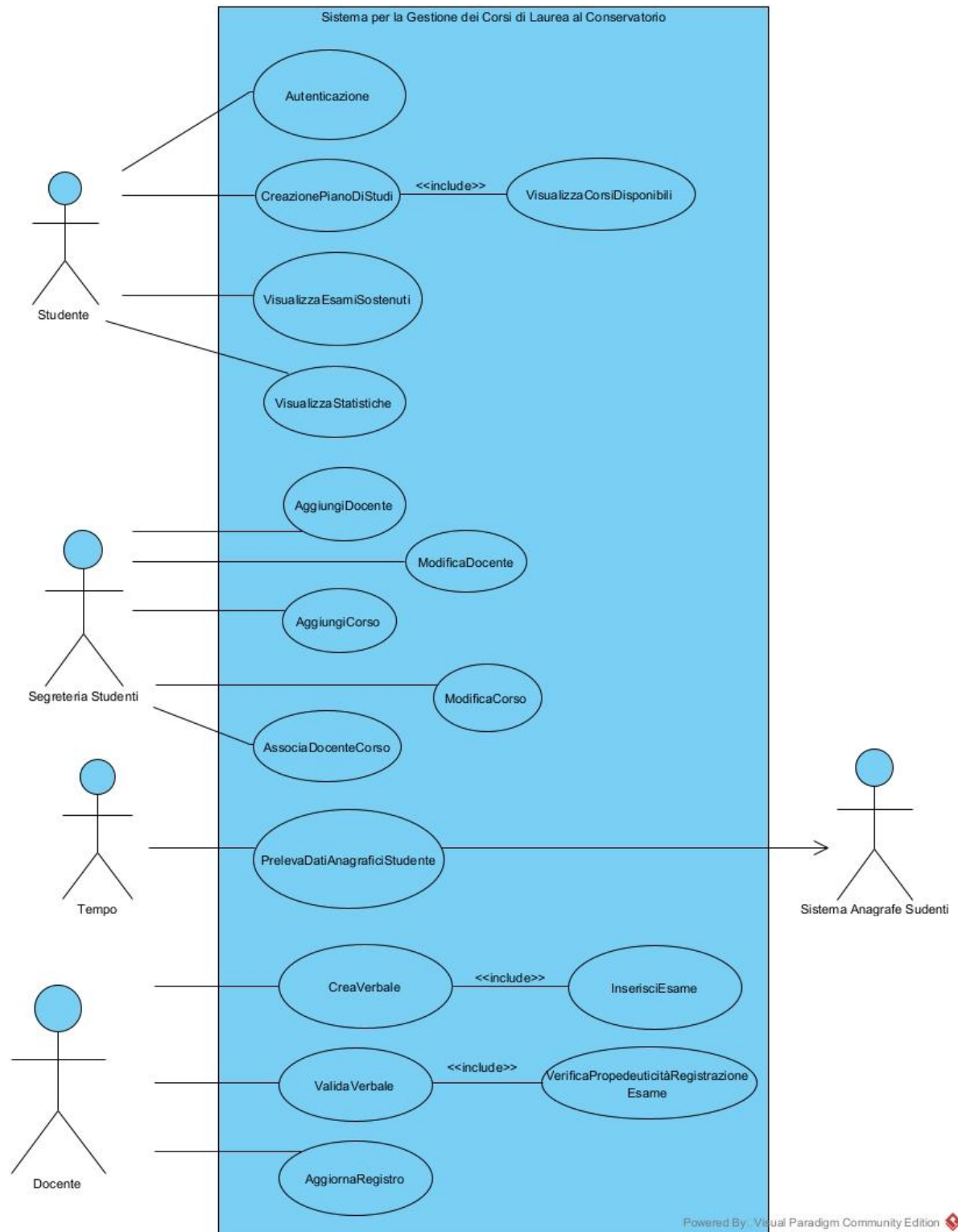
#### Casi d'uso:

- **UC01:** Autenticazione
- **UC02:** VisualizzaCorsiDisponibili
- **UC03:** CreazionePianoDiStudi
- **UC04:** VisualizzaEsamiSostenuti
- **UC05:** VisualizzaStatistiche
- **UC06:** AggiungiDocente
- **UC07:** AggiungiCorso
- **UC08:** ModificaDocente
- **UC09:** ModificaCorso
- **UC10:** AssociaDocente\_Corso
- **UC11:** PrelevaDatiAnagraficiStudente
- **UC12:** CreaVerbale
- **UC13:** InserisciEsame
- **UC14:** ValidaVerbale
- **UC15:** VerificaPropedeuticitàRegistrazioneEsame
- **UC16:** AggiornaRegistro

## CASI D'USO

<u>Caso d'uso</u>	<u>Attore Primario</u>	<u>Attore Secondario</u>	<u>CDU inclusione</u>
<b>UC01: Autenticazione</b>	Studente	-	-
<b>UC02: VisualizzaCorsiDisponibili</b>	Studente	-	
<b>UC03: CreazionePianoDiStudi</b>	Studente	-	UC02
<b>UC04: VisualizzaEsamiSostenuti</b>	Studente	-	
<b>UC05: VisualizzaStatistiche</b>	Studente	-	
<b>UC06: AggiungiDocente</b>	Segreteria	-	
<b>UC07: AggiungiCorso</b>	Segreteria	-	
<b>UC08: ModificaDocente</b>	Segreteria	-	
<b>UC09: ModificaCorso</b>	Segreteria	-	
<b>UC10: AssociaDocente_Corso</b>	Segreteria		
<b>UC11: PrelevaDatiAnagraficiStudente</b>	Tempo	Sistema Anagrafe Studente	
<b>UC12: CreaVerbale</b>	Docente	-	UC13
<b>UC13: InserisciEsame</b>	Docente	-	
<b>UC14: ValidaVerbale</b>	Docente	-	
<b>UC15: VerificaPropedeuticitàRegistrazioneEsame</b>	Docente	-	UC14
<b>UC16: AggiornaRegistro</b>	Docente	-	

## 2.5.2 Diagramma dei casi d'uso



### 2.5.3 Scenari

Caso d'uso:	Autenticazione
<b>Attore primario</b> <b>Attore secondario</b> <b>Descrizione</b> <b>Pre-Condizioni</b> <b>Sequenza di eventi principale</b>	Studente - Uno studente esegue l'autenticazione per accedere ai servizi del portale. Lo studente deve essere registrato. 1. Il caso d'uso inizia quando lo studente tenta di accedere al portale. 2. Lo studente inserisce username e password. 3. Il sistema verifica la correttezza dei dati inseriti. 4. <b>if</b> i dati inseriti sono corretti 4.1. Lo studente accede al sistema
<b>Post-Condizioni</b> <b>Casi d'uso correlati</b> <b>Sequenza di eventi alternativi</b>	Lo studente accede al sistema. - Al punto 4, se i dati inseriti non corrispondono con quelli registrati il sistema mostra a video messaggio di errore e richiede l'inserimento i dati.

Caso d'uso:	VisualizzaCorsiDisponibili
<b>Attore primario</b>	Studente
<b>Attore secondario</b>	-
<b>Descrizione</b>	Lo studente richiede al sistema di visualizzare tutti i corsi disponibili
<b>Pre-Condizioni</b>	Lo studente deve essere autenticato.
<b>Sequenza di eventi principale</b>	1 Il sistema mostra a video tutte le operazioni che lo studente può effettuare. 2. Lo studente sceglie di visualizzare i corsi disponibili. 3. Il sistema mostra i corsi disponibili.
<b>Post-Condizioni</b>	-
<b>Casi d'uso correlati</b>	<i>Autenticazione</i>
<b>Sequenza di eventi alternativi</b>	Nel punto 3, se non ci sono corsi disponibili, il sistema mostra un messaggio di errore.

<b>Caso d'uso:</b>	<b>CreazionePianoDiStudi</b>
<b>Attore primario</b>	Studente
<b>Attore secondario</b>	-
<b>Descrizione</b>	Lo studente accede al sistema per poter creare il suo piano di studi personale
<b>Pre-Condizioni</b>	Lo studente deve essere autenticato.
<b>Sequenza di eventi principale</b>	<ol style="list-style-type: none"> <li>1. Il caso d'uso inizia quando lo studente seleziona l'opzione relativa alla creazione di un piano di studi personalizzato.</li> <li>2. Include &lt;&lt;<i>VisualizzaCorsiDisponibili</i>&gt;&gt;</li> <li>3. Lo studente seleziona ed aggiunge un corso nel nuovo piano di studi. <ol style="list-style-type: none"> <li>3.1 <b>if</b> il corso è stato già aggiunto, il sistema restituisce un messaggio di errore.</li> </ol> </li> <li>4. il sistema verifica che il numero di cfu aggiunti sia pari al valore standard.</li> <li>5. Il sistema mostra il piano di studi creato in seguito ad un messaggio di avvenuta creazione.</li> </ol>
<b>Post-Condizioni</b>	È stato creato un piano di studi
<b>Casi d'uso correlati</b>	<i>VisualizzaCorsiDisponibili</i>
<b>Sequenza di eventi alternativi</b>	Se al punto 4 il numero di cfu è minore del valore minimo richiesto, il sistema restituisce un messaggio di errore.

<b>Caso d'uso:</b>	<b>AggiungiDocente</b>
<b>Attore primario</b>	Segreteria Studenti
<b>Attore secondario</b>	-
<b>Descrizione</b>	Un operatore della Segreteria Studenti inserisce un nuovo docente nel sistema.
<b>Pre-Condizioni</b>	Il docente non deve essere già registrato nel sistema.
<b>Sequenza di eventi principale</b>	<ol style="list-style-type: none"> <li>1. Il caso d'uso inizia quando l'operatore della Segreteria accede alla sezione "Gestione Docenti".</li> <li>2. Il sistema mostra l'interfaccia per l'aggiunta di un nuovo docente.</li> <li>3. L'operatore inserisce il Nome, Cognome e Password del docente.</li> <li>4. Il sistema verifica che nome e cognome siano composti da soli caratteri dell' alfabeto.</li> <li>5. Il sistema verifica che la password sia esattamente di 10 caratteri alfanumerici.</li> <li>6. Il sistema crea una matricola per il docente e verifica che non esista già un docente con la stessa matricola.</li> <li>7. <b>if</b> non è stato trovato nessun docente con la medesima matricola: Il sistema aggiunge i dati relativi al docente.</li> <li>8. Il sistema mostra un messaggio di conferma di aggiunta del docente.</li> <li>9. Il caso d'uso termina.</li> </ol>
<b>Post-Condizioni</b>	Un docente è stato aggiunto.
<b>Casi d'uso correlati</b>	<i>ModificaDocente</i> .

<b>Sequenza di eventi alternativi</b>	<p>Al punto 4, se esiste già una matricola corrispondente a quella che si sta registrando, il sistema mostra un messaggio di errore e annulla l'operazione, tornando al punto 3.</p> <p>Al punto 3, se i dati risultano incompleti, il sistema invia una notifica di avvertimento e aspetta il completamento dei dati.</p> <p>Al punto 4, se il nome o il cognome inseriti possiedono caratteri speciali o numeri, il sistema mostra un messaggio di errore e annulla l'operazione, tornando al punto 3.</p> <p>Al punto 5, se la password non contiene esattamente 10 caratteri o se uno o più caratteri non sono alfanumerici, il sistema mostra un messaggio di errore e annulla l'operazione, tornando al punto 3.</p>
---------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<b>Caso d'uso:</b>	<b>CreaVerbale</b>
<b>Attore primario</b>	Docente
<b>Attore secondario</b>	-
<b>Descrizione</b>	Il docente crea un nuovo verbale.
<b>Pre-Condizioni</b>	Il docente è autenticato al sistema.
<b>Sequenza di eventi principale</b>	<ol style="list-style-type: none"> <li>1. Il caso d'uso inizia quando il docente seleziona l'opzione relativa alla creazione verbale.</li> <li>2. Il docente crea il verbale dell'esame inserendo data e la propria matricola.</li> <li>3. Il sistema verifica che la matricola inserita sia associata ad un docente.</li> <li>4. Il sistema mostra a video tutti i corsi che il docente ha assegnati relativi alla data inserita sul verbale.</li> <li>5. Il docente seleziona il corso al quale attribuire il verbale.</li> <li>6. Il sistema mostra a video i dati del verbale appena creato.</li> <li>7. Il docente ora può inserire gli esami nel verbale.</li> <li>8. <b>while:</b>  &lt;&lt;include InserisciEsame&gt;&gt; </li> </ol>
<b>Post-Condizioni</b>	Un verbale è stato memorizzato.
<b>Casi d'uso correlati</b>	
<b>Sequenza di eventi alternativi</b>	<p>Al punto 3, se non è presente nessuna matricola nel database viene restituito un messaggio d'errore.</p> <p>Al punto 4 se non ci sono corsi assegnati al docente per quell'anno accademico viene restituito un messaggio d'errore.</p>

Caso d'uso:	InserisciEsame
<p><b>Attore primario</b></p> <p><b>Attore secondario</b></p> <p><b>Descrizione</b></p> <p><b>Pre-Condizioni</b></p> <p><b>Sequenza di eventi principale</b></p> <p><b>Post-Condizioni</b></p> <p><b>Casi d'uso correlati</b></p> <p><b>Sequenza di eventi alternativi</b></p>	<p>Docente</p> <p>-</p> <p>Il docente inserisce un nuovo esame.          é stato creato un verbale.</p> <ol style="list-style-type: none"> <li>1. Il docente inserisce la matricola dello studente che ha sostenuto l'esame, la valutazione (HP lode) e le note.</li> <li>2. Il sistema verifica che esista uno studente con la matricola inserita.</li> <li>3. Il sistema verifica che il voto sia compreso tra 18 e 30 e che non venga assegnata la lode ad un voto diverso da 30.</li> <li>4. Il docente conferma l'inserimento dei dati</li> <li>5. Il sistema mostra a video un messaggio di conferma inserimento esame</li> <li>6. Il sistema permetta al docente di inserire un altro esame in quel verbale o di tornare indietro.</li> </ol> <p>Un esame è stato aggiunto al verbale</p> <p>-</p> <p>Al punto 2 se il controllo fallisce viene restituito un messaggio d'errore e richiesto di nuovo l'inserimento dei dati.</p> <p>Al punto 3 se il controllo fallisce viene restituito un messaggio d'errore e richiesto di nuovo l'inserimento dei dati.</p>

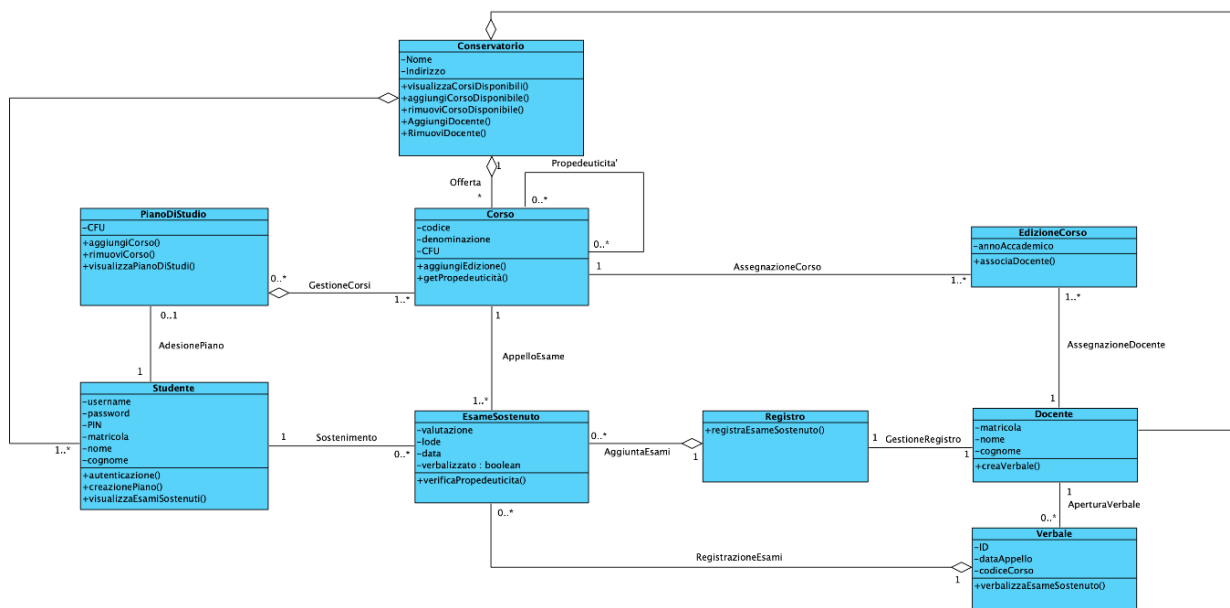


## 2.6 Diagramma delle classi

Di seguito è riportato il diagramma delle classi di analisi corrispondente al System Domain Model. Esso rappresenta le principali entità concettuali del dominio e le loro responsabilità in relazione ai requisiti funzionali.

Le classi modellate non corrispondono necessariamente a quelle che saranno implementate, ma aiutano a comprendere la struttura logica e le funzionalità richieste.

Le operazioni riportate hanno nomi generici ed esplicativi. Inoltre, tali operazioni sono limitate a quelle significative dal punto di vista dell'utente o del dominio, tralasciando dettagli implementativi.



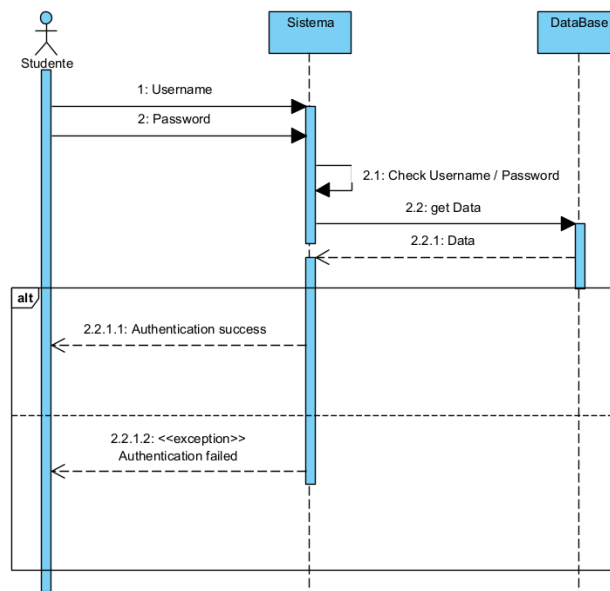
## 2.6.1 Responsabilità

RESPONSABILITA'	CLASSE
Autenticazione	Conservatorio
CreazionePiano	Studente
CreaVerbale	Docente
VisualizzaPianoDiStudi	PianoDiStudio
AggiungiCorso	PianoDiStudio
ModificaCorso	PianoDiStudio
VerificaPropedeuticità	Verbale
VerbalizzaEsameSostenuto	Verbale
RegistraEsameSostenuto	Registro
AggiungiEdizione	CorsoDiStudio
VisualizzaEsamiSostenuti	EsameSostenuto
AggiungiCorsoDisponibile	Conservatorio
VisualizzaCorsiDisponibili	Conservatorio
RimuoviCorsoDisponibile	Conservatorio
AggiungiDocente	Conservatorio
ModificaDocente	Conservatorio
RimuoviDocente	Conservatorio
AssociaDocente	EdizioneCorso

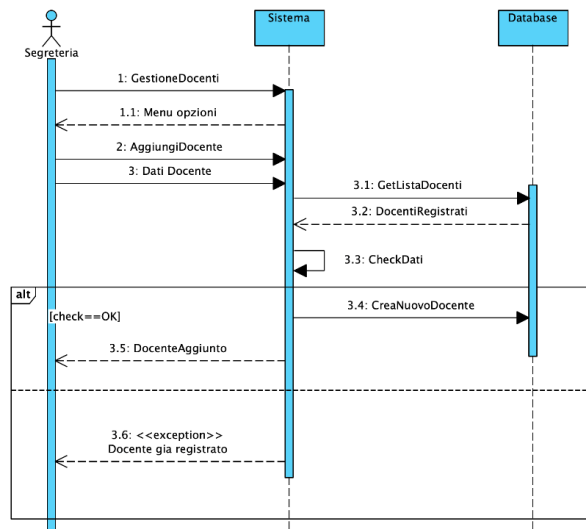
## 2.7 Sequence Diagrams

Si riportano alcuni esempi di diagrammi di sequenza di analisi per i casi d'uso precedentemente individuati.

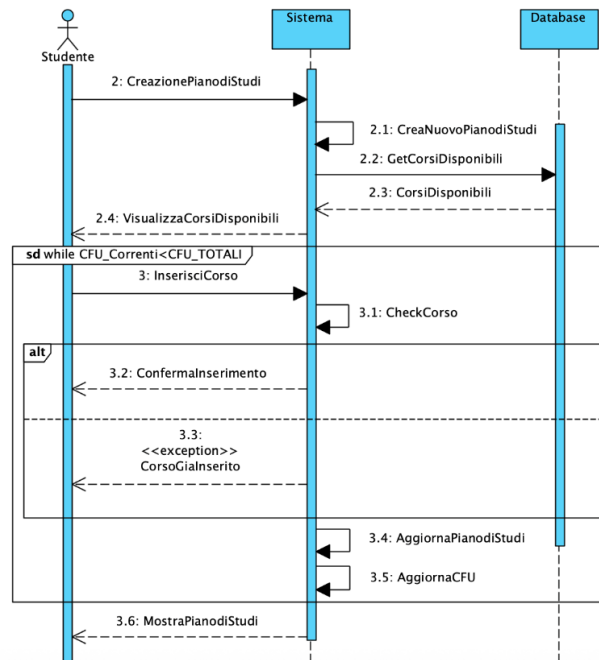
### Autenticazione



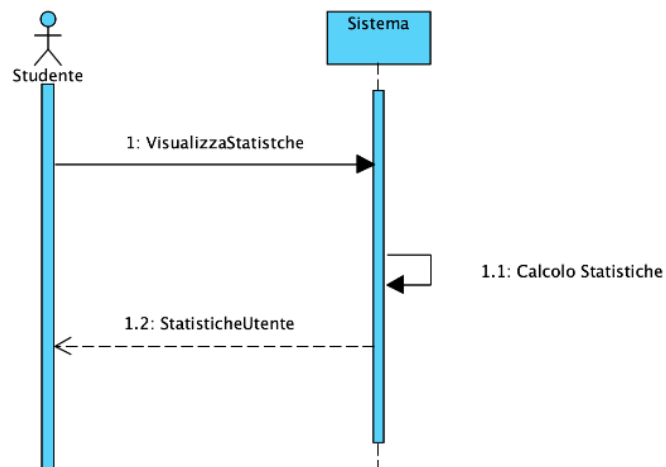
### AggiungiDocente



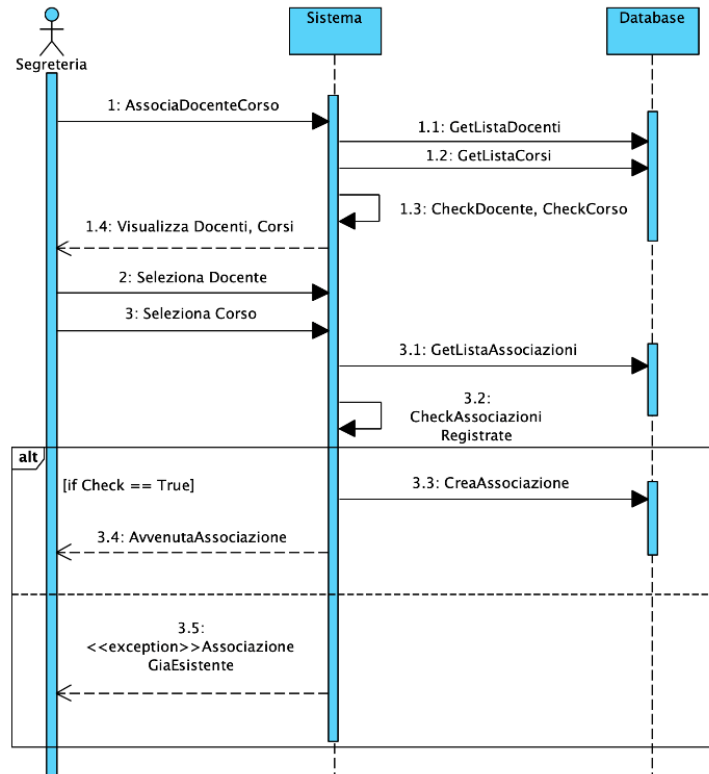
## CreazionePianoDiStudi



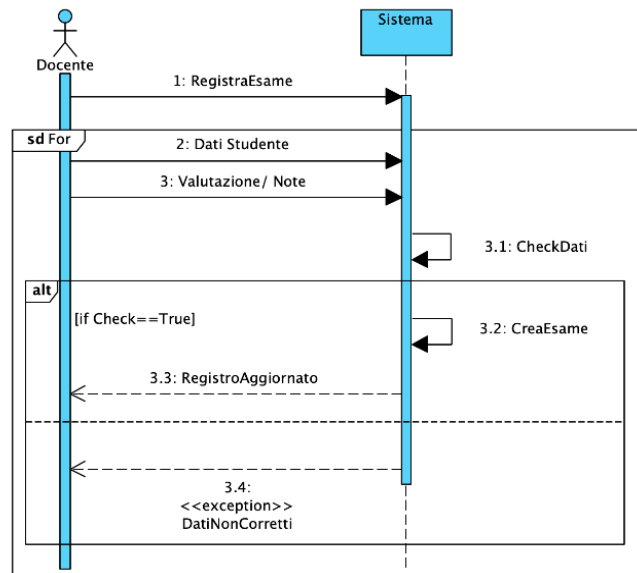
## VisualizzaStatistiche



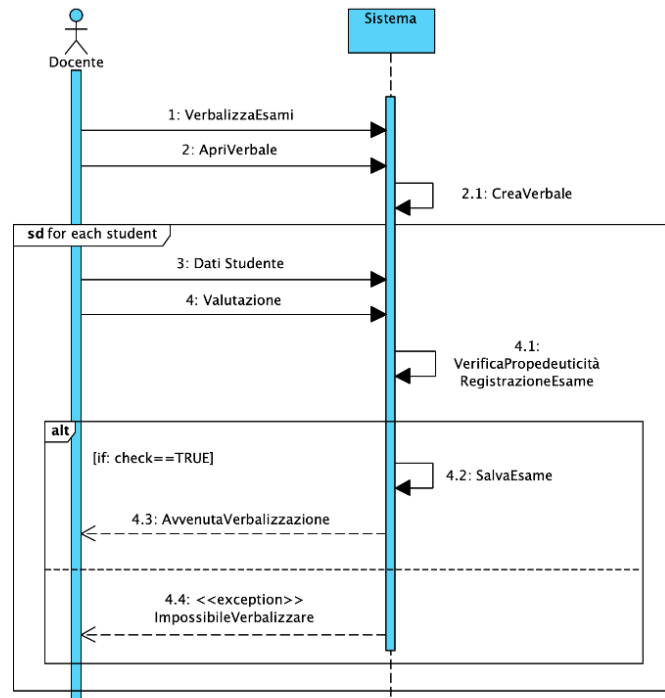
## AssociaDocenteCorso



## RegistraEsame



## VerbalizzaEsami



## 3. Piano di test funzionale

Si progettano i casi di test funzionale facendo uso della tecnica Category Partition Testing per le varie funzionalità, descrivendo il procedimento di calcolo.

### 3.1 Autenticazione

AUTENTICAZIONE	
USERNAME	PASSWORD
<ul style="list-style-type: none"><li>• Stringa di caratteri</li><li>• Stringa di caratteri vuota <b>[ERROR]</b></li><li>• Stringa di caratteri non presente sul database <b>[ERROR]</b></li></ul>	<ul style="list-style-type: none"><li>• Stringa di caratteri</li><li>• Stringa di caratteri vuota <b>[ERROR]</b></li><li>• Stringa di caratteri non presente sul database <b>[ERROR]</b></li></ul>

Il numero di test da effettuare senza particolari vincoli è:  $3 \times 3 = 9$ .

Considerando i vincoli [ERROR], il numero di test da effettuare per testare singolarmente ogni vincolo è  $2(\text{Username}) + 2(\text{Password}) = 4$ .

il n. di test che ne risulta è:  $(1 \times 1) + 4 = 5$ .

TEST SUITE DI AUTENTICAZIONE						
Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output attesi	Post-condizioni attese
1	Autenticazione avvenuta con successo	Username esistente, Password corretta	Lo studente esiste nel DB con credenziali valide	(studente1, password123)	Autenticazione riuscita	L'utente viene autenticato e accede all'

TEST SUITE DI AUTENTICAZIONE						
						interfaccia studente
2	Autenticazio ne fallita: Username non esistente	Username non esistente [ERROR]	Lo studente NON esiste nel DB	(studenteINESI STENTE, qualsiasiPassw ord)	"Errore di autenticazione: Username non esistente."	L'utente non viene autenticato; viene mostrato un messaggio di errore.
3	Autenticazio ne fallita: Password errata	Username esistente, Password non corretta [ERROR]	Lo studente esiste nel DB con username ma con password diversa	(studente1, passwordERRA TA)	"Errore di autenticazione: Password non corretta."	L'utente non viene autenticato; viene mostrato un messaggio di errore.
4	Autenticazio ne fallita: Username vuoto	Username vuoto [ERROR]	Nessuna	("", password123)	"Errore di autenticazione: Riempire tutti i campi."	L'utente non viene autenticato; viene mostrato un messaggio di errore.
5	Autenticazio ne fallita: Password vuota	Password vuota [ERROR]	Nessuna	(studente1, "")	"Errore di autenticazione: Riempire tutti i campi."	L'utente non viene autenticato; viene mostrato un messaggio di errore.



TEST SUITE DI AUTENTICAZIONE						
6	Autenticazione fallita: Username e Password vuoti	Username vuoto [ERROR], Password vuota [ERROR]	Nessuna	("", "")	"Errore di autenticazione: Riempire tutti i campi."	L'utente non viene autenticato; viene mostrato un messaggio di errore.
7	Autenticazione fallita: Errore del database	Errore interno DAO [ERROR]	Errore di connessione e/query al DB	(qualsiasiUsername, qualsiasiPassword)	"Si è verificato un errore inatteso. Contatta l'amministratore."	L'utente non viene autenticato; viene mostrato un messaggio di errore.

### 3.2 InserisciVerbale

InserisciVerbale		
Matricola	Data	Corsi Associati
<ul style="list-style-type: none"> <li>• Stringa lunghezza =4</li> <li>• Stringa lunghezza &lt;4 [ERROR]</li> <li>• Stringa lunghezza &gt;4 [ERROR]</li> <li>• Stringa vuota [ERROR]</li> <li>• Stringa con simboli diversi dai caratteri [ERROR]</li> <li>• Stringa non presente nel database [ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>• Stringa nel formato corretto</li> <li>• Stringa con simboli diversi dai caratteri [ERROR]</li> <li>• Stringa data successiva alla data corrente [ERROR]</li> <li>• Stringa vuota [ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>• Esiste almeno un corso associato alla matricola del docente per l'a.a. specificato.</li> <li>• Non ci sono corsi associati alla matricola docente per l'a.a. specificato. [ERROR]</li> <li>• Il docente non è mai stato associato a corsi di studio [ERROR]</li> </ul>

Casi  $6 \times 4 \times 3 = 72$

Con i vincoli [ERROR], invece, il numero di test da eseguire per testare singolarmente i vincoli è 10: (5 per la matricola, 3 per la data e 2 per il corso di studio).

TEST SUITE di INSERISCI VERBALE						
Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output attesi	Post-condizioni attese
1	Tutti input validi, corsi associati	Matricola valida, Data valida, Corsi associati	Esiste almeno un corso di studio associato al docente	{matricola : D001, data: 14/07/2024 }	Verbale creato correttamente	Creazione di un verbale associato al docente, in data e relativo a quel corso.
2	Matricola formato errato	Matricola non valida (formato errato)		{matricola : D01, data: 14/07/2024 }	Errore formato matricola	Nessun verbale creato
3	Data vuota	Data mancante (campo obbligatorio)		{matricola : D001, data: null}	Campo data obbligatorio	Nessun verbale creato
4	Matricola vuota	Matricola mancante (campo obbligatorio)		{{matricola : null, data: 14/07/2024 }	Campo matricola obbligatorio	Nessun verbale creato
5	Data in formato errato (aaaa/mm/gg)	Data non valida (formato errato)		{matricola : D001, data: 2024/07/14 }	Formato data non valido	Nessun verbale creato
6	Data futura	Data non valida (data futura)		{matricola : D001, data: 14/07/2030 }	Non possiamo ancora predire il futuro	Nessun verbale creato
7	Matricola formato corretto ma docente non trovato	Matricola valida, docente inesistente	D999 non presente nel database	{matricola : D999, data: 14/07/2024 }	Matricola docente non trovata	Nessun verbale creato
8	Docente esiste ma nessun corso associato	Matricola valida, Data valida, corsi non associati	D001 esiste ma non ha corsi	{matricola : D001, data: 14/07/2024 }	Non ci sono corsi associati al docente	Nessun verbale creato
9	Entrambi i campi vuoti	Matricola mancante, Data mancante		{matricola : null, data: null}	Campo matricola e campo data obbligatori	Nessun verbale creato
10	Matricola con simboli	Matricola non valida (caratteri speciali)		{matricola : D0!1, data: 14/07/2024 }	Formato matricola non valido	Nessun verbale creato

11	Data con caratteri non numerici	Data non valida (caratteri non numerici)		{matricola : D001, data: 14/1u/2024 }	Formato data non valido	Nessun verbale creato
12	Data formattata bene ma non esistente (es: 31/02/2024)	Data non valida (data inesistente)		{matricola : D001, data: 31/02/2024 }	Formato data non valido	Nessun verbale creato

### 3.3 Aggiungi Docente

AggiungiDocente		
Nome	Cognome	Password
<ul style="list-style-type: none"> <li>• Stringa di caratteri di lunghezza <math>\leq 20</math></li> <li>• Stringa di caratteri di lunghezza <math>&gt; 20</math> <b>[ERROR]</b></li> <li>• Stringa che contiene simboli che non sono caratteri <b>[ERROR]</b></li> </ul>	<ul style="list-style-type: none"> <li>• Stringa di caratteri di lunghezza <math>\leq 20</math></li> <li>• Stringa di caratteri di lunghezza <math>&gt; 20</math> <b>[ERROR]</b></li> <li>• Stringa che contiene simboli che non sono caratteri <b>[ERROR]</b></li> </ul>	<ul style="list-style-type: none"> <li>• Stringa di caratteri alfanumerici di lunghezza = 10</li> <li>• Stringa di caratteri alfanumerici di lunghezza <math>&gt; 10</math> <b>[ERROR]</b></li> <li>• Stringa di caratteri alfanumerici di lunghezza <math>&lt; 10</math> <b>[ERROR]</b></li> <li>• Stringa contenente caratteri speciali <b>[ERROR]</b></li> </ul>

Il numero di test da effettuare senza particolari vincoli è:  $3 \times 3 \times 4 = 36$ .

Considerando i vincoli **[ERROR]**, il numero di test da effettuare per testare singolarmente ogni vincolo è  $2(\text{Nome}) + 2(\text{Cognome}) + 3(\text{Password}) = 7$ .

il n. di test che ne risulta è:  $(1*1*1) + 7 = 8$ .



### 3.4 CreaPianoDiStudio

SalvaPianoDiStudi	
Corsi Selezionati	IdPiano
Corsi Selezionati è nullo[ERROR]	IdPiano è negativo [ERROR]
Corsi Selezionati è vuoto[ERROR]	IdPiano è nullo[ERROR]
Corsi Selezionati contiene almeno un corso	IdPiano è un intero positivo

Il numero di test da effettuare senza particolari vincoli è:  $3 \times 3 = 9$ .

Considerando i vincoli [ERROR], il numero di test da effettuare per testare singolarmente ogni vincolo è  $2(\text{corsiSelezionati}) + 2(\text{idpiano}) = 4$ .

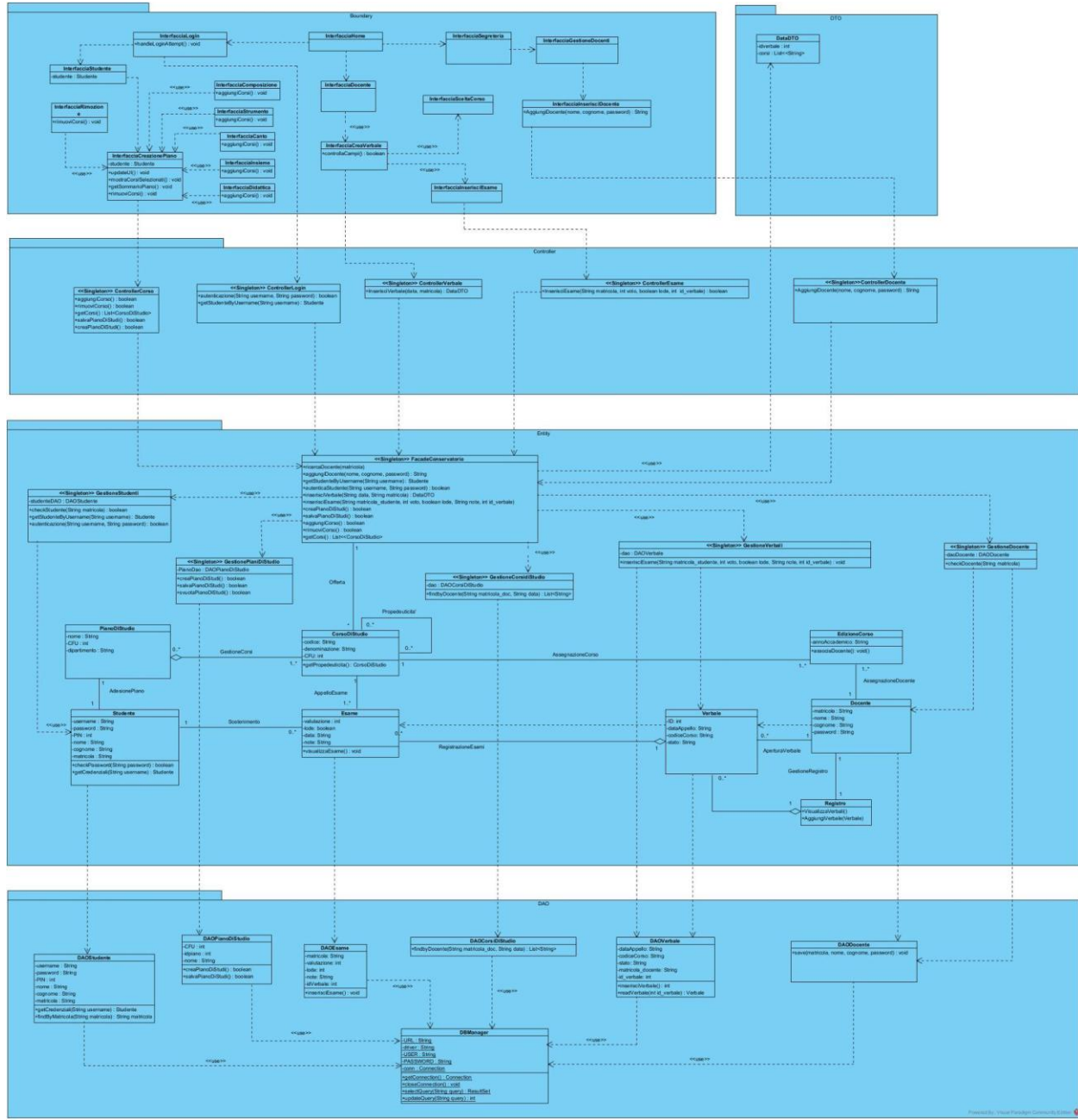
il n. di test che ne risulta è:  $(1 \times 1) + 4 = 5$ .

TEST CreaPianoDiStudio						
Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output attesi	Post-condizioni attese
1	Corsi Selezionati è nullo	Corsi Selezionati non è nullo	Nessuna	idpiano = 1, corsiSelezionati = null	False	Nessuna interazione con DBManager. Messaggio "Nessun corso da salvare." nel log.
2	corsiSelezionati è una lista vuota	Username non esistente [ERROR]	Nessuna	idpiano = 1, corsiSelezionati = new ArrayList<>()	false	Nessuna interazione con DBManager. Messaggio "Nessun corso da salvare." nel log.
3	idpiano è zero	Ramo if (idpiano <= 0)	corsiSelezionati non nullo e non vuoto.	idpiano = 0, corsiSelezionati = [CorsoDiStudio (codice=101)]	false	Nessuna interazione con DBManager. Messaggio "ID piano non valido." nel log.

4	idpiano è negativo	Ramo if (idpiano <= 0)	corsiSelezionati non nullo e non vuoto.	idpiano = -5, corsiSelezionati = [CorsoDiStudio (codice=101)]	false	Nessuna interazione con DBManager. Messaggio "ID piano non valido." nel log.
5	Salvataggio di corsi sul database (successo)	Percorso principale, ciclo for	DB in ascolto	idpiano = 10, corsiSelezionati = [CorsoDiStudio (codice=201)]	true	DBManager.updateQuery() chiamato 1 volta con INSERT INTO corsiselezionati (idpiano, idcorso) VALUES (10, 201).
6	SQLException durante il salvataggio	Blocco catch (SQLException)	DB in ascolto	idpiano = 30, corsiSelezionati = [CorsoDiStudio (codice=401)]	false	Messaggio di errore SQL nel log.
7	ClassNotFoundException durante il salvataggio	Blocco catch (ClassNotFoundException)	DB in ascolto	idpiano = 40, corsiSelezionati = [CorsoDiStudio (codice=501)]	false	Messaggio "Classe non trovata nel database" nel log.

# 4. Progettazione

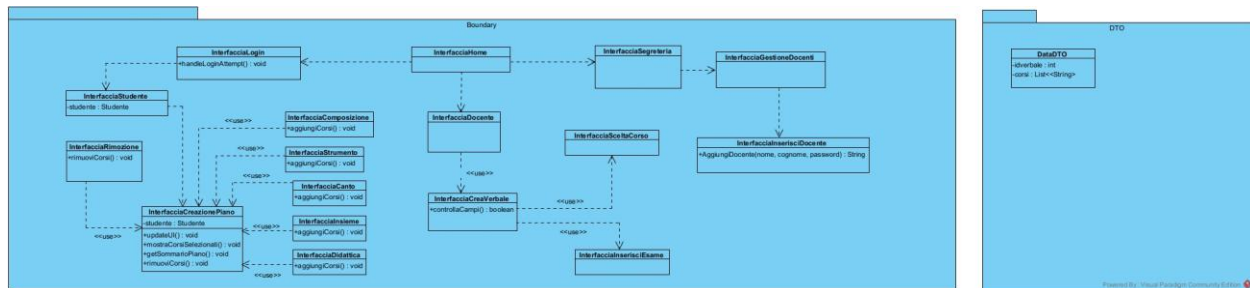
## 4.1: Diagramma delle classi (modello BCED)



### 4.1.2: Pattern BCED

#### 4.1.2.1 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.



#### 4.1.2.2 Package Controller

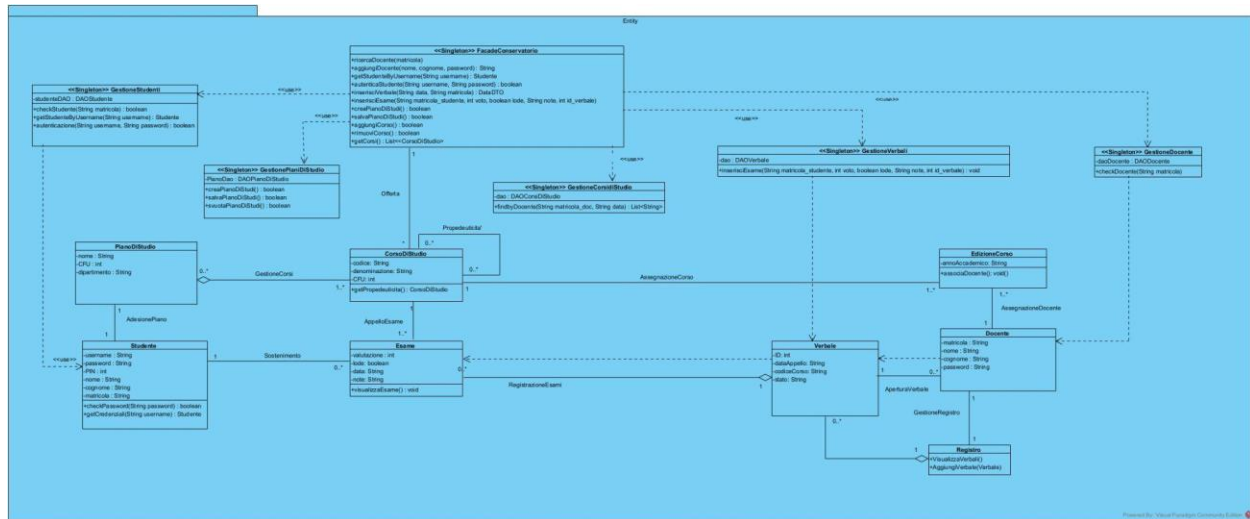
Questo package contiene gli oggetti che percepiscono gli eventi generati dalle interazioni con l'interfaccia utente e ne demandano la gestione all'unico componente del sistema software responsabile della gestione della Business Logic, il package Entity.





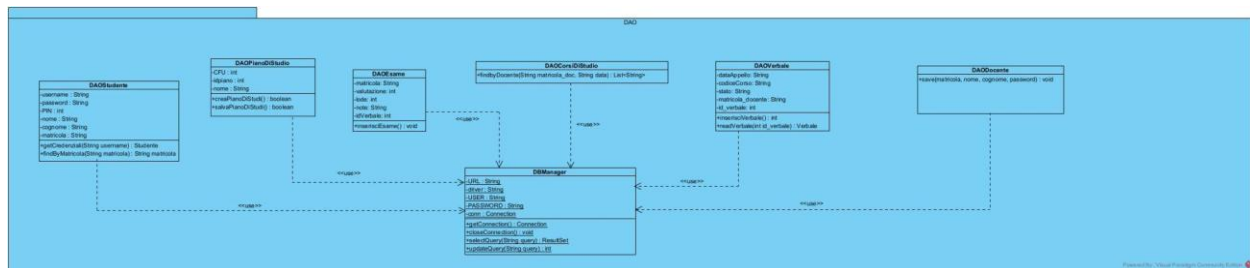
#### 4.1.2.3 Package Entity

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.

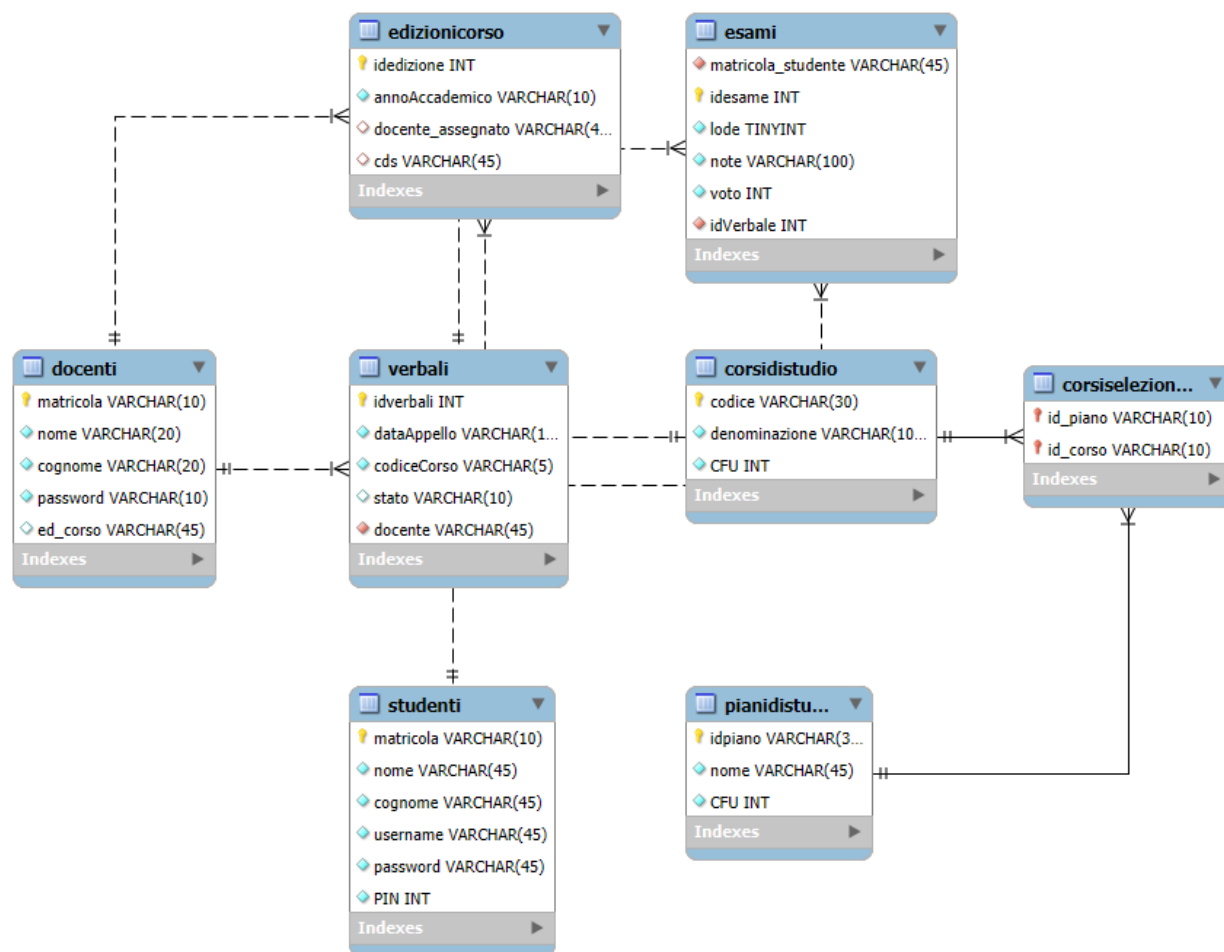


#### 4.1.2.4 Package Database

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, DBManager, che costituisce di fatto l'unico punto di accesso vero e proprio al DB, sfruttando i metodi che questa mette a disposizione.



## Scelte progettuali del Database



## Relazioni di tipo multi-a-molti

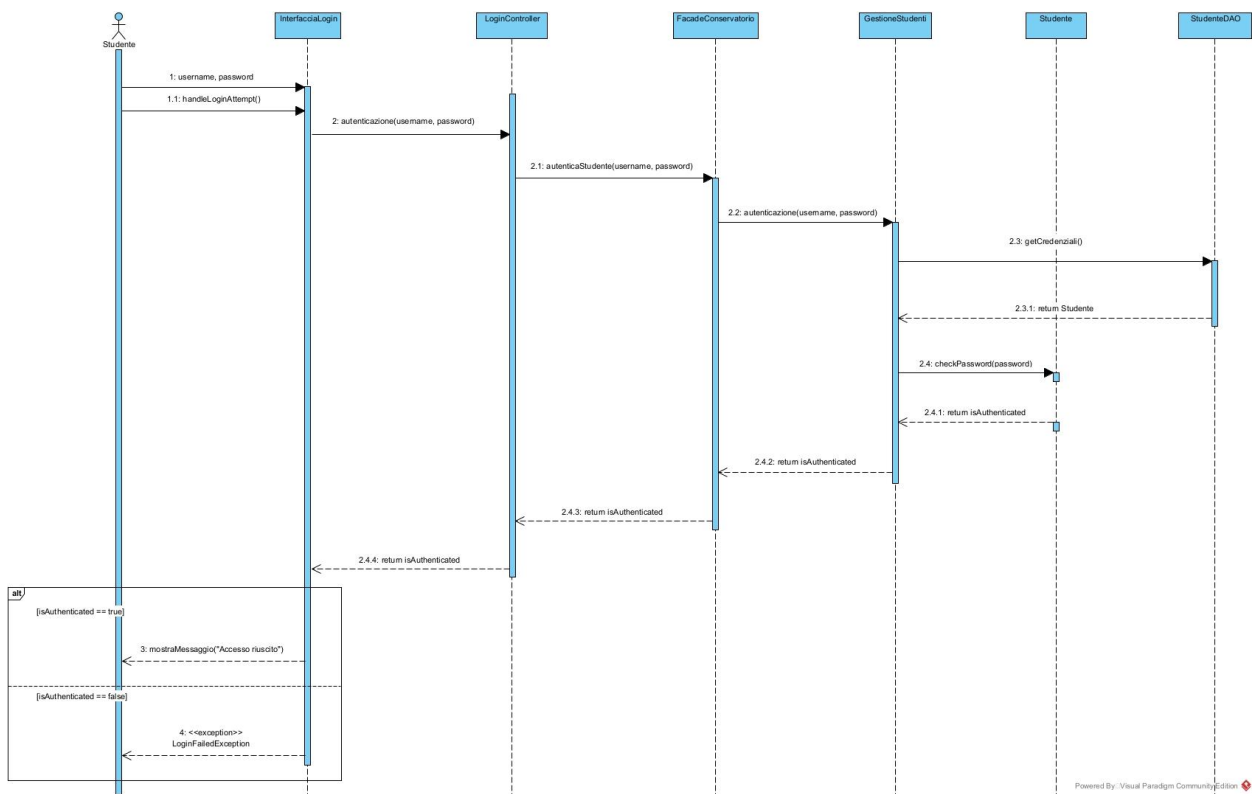
Le relazioni complesse tra entità vengono gestite mediante l'utilizzo di tabelle associative, le quali consentono di modellare in maniera appropriata i legami di tipo multi-a-molti, permettendo al contempo di arricchire le informazioni ad esse associate.

- **verbi**: è relativo al corso di studio e collega il docente di quella edizione del corso di studio. Documenta un appello d'esame, specificando data e stato. Non presenta una effettiva relazione multi-a-molti ma 2 relazioni multi-a-uno (considerabile come una tabella associativa parzialmente).
- **corsiselezionati**: Indica quali corsi sono stati selezionati per ciascun piano di studio.
- **esami**: Tiene traccia degli esami sostenuti dagli studenti, andando ad associare ogni esame con un'edizione del corso.

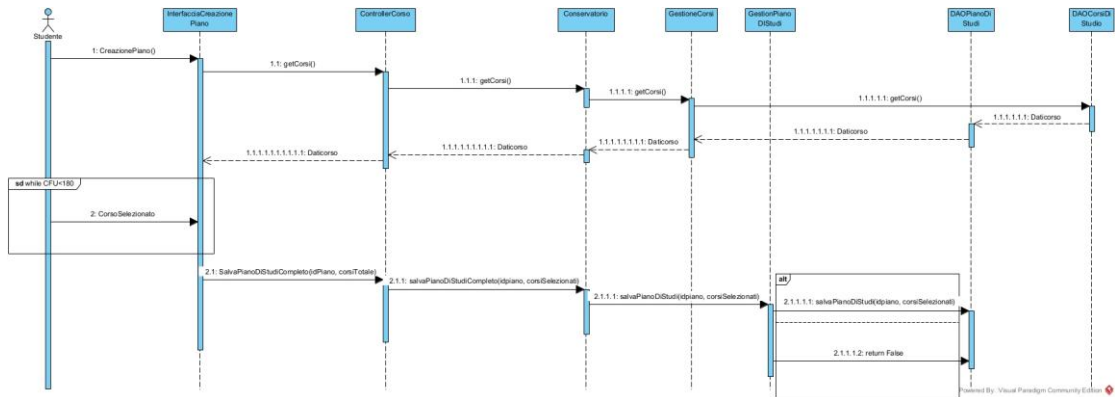
## 4.2 Diagrammi di sequenza

Si riportano di seguito i diagrammi di sequenza di progetto per i casi d'uso sviluppati fino alla codifica in Java.

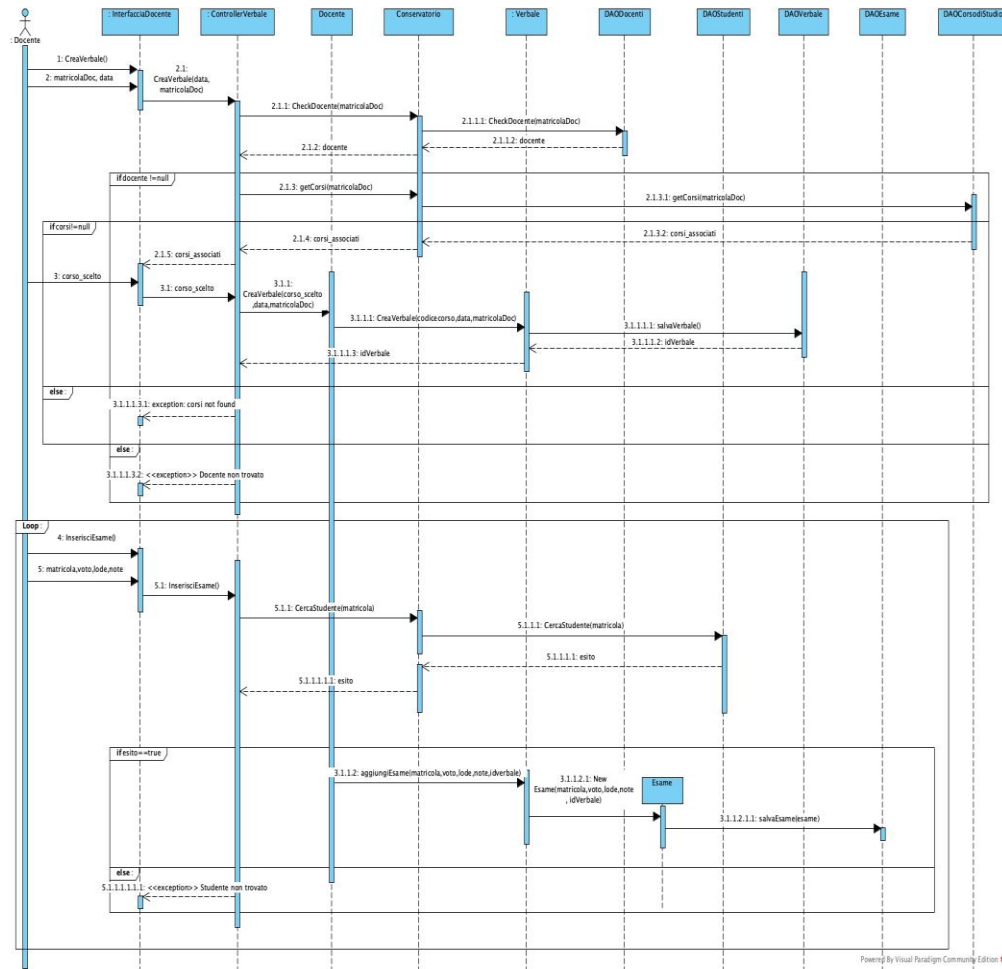
### 4.2.1 Autenticazione



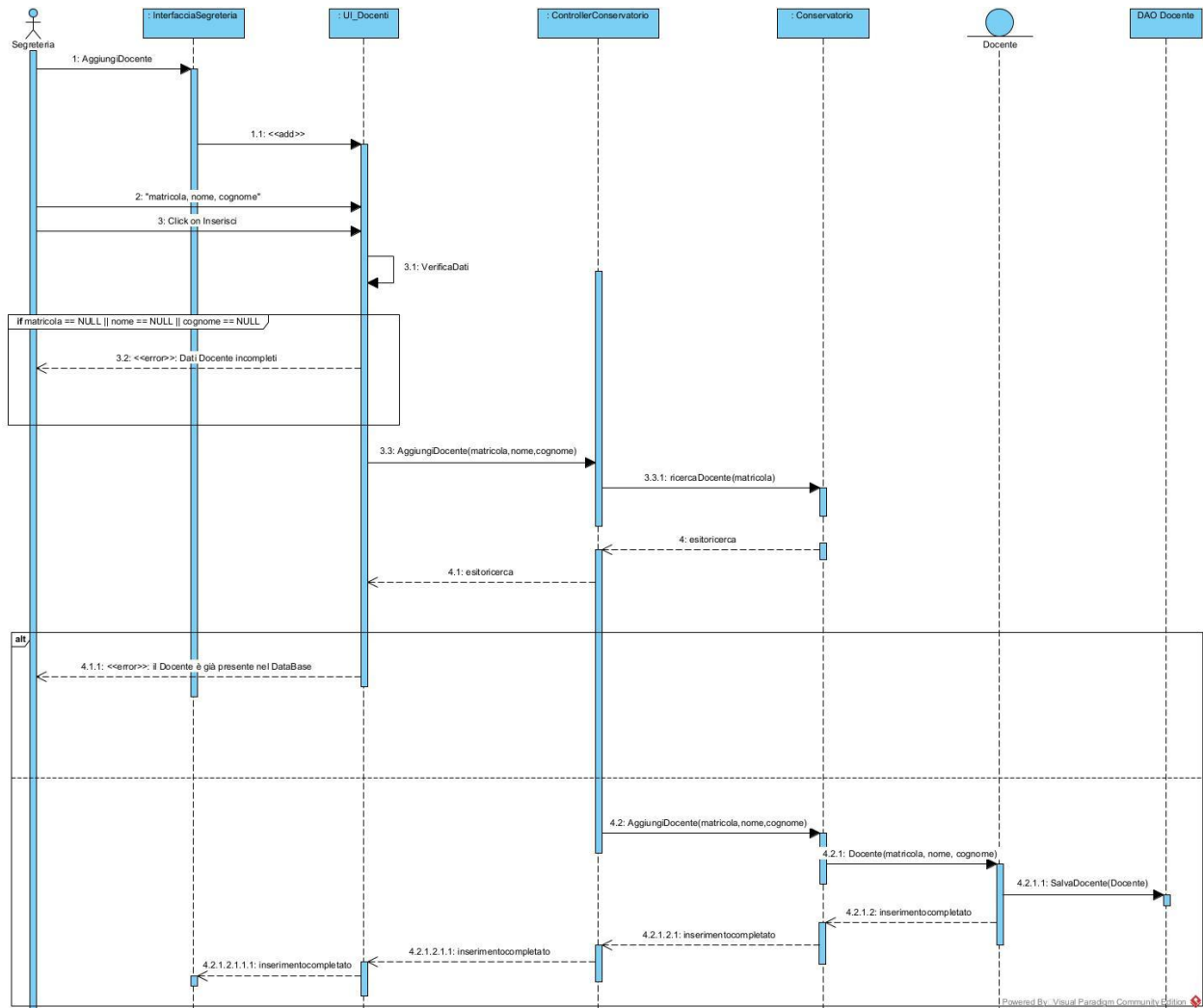
#### 4.2.2 CreaPianoDiStudi



### 4.2.3 VerbalizzaEsame



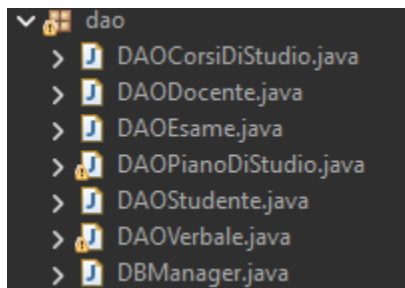
## 4.2.4 AggiungiDocente



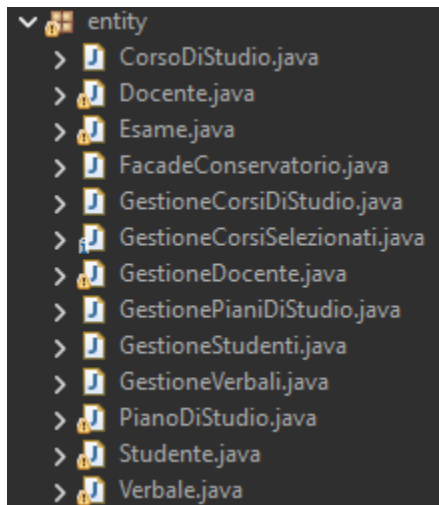
## 5. Implementazione

Abbiamo sviluppato l'applicazione in Java, utilizzando Eclipse IDE come ambiente di sviluppo. La scelta di Java ci ha permesso di sfruttare la sua elevata portabilità, rendendo l'applicazione eseguibile su qualsiasi sistema dotato di Java Virtual Machine, senza modifiche al codice. Inoltre, l'uso di Eclipse ha reso più efficiente la scrittura, l'organizzazione e il debugging, facilitando lo sviluppo all'interno di un'architettura modulare e chiara. In generale, il nostro team di sviluppatori si è impegnato a seguire le best practice dell'Ingegneria del Software, adottando l'architettura a strati BCED (Boundary, Controller, Entity, DAO) e garantendo un rigoroso rispetto del principio di comunicazione tra layer adiacenti. Per facilitare il passaggio dei dati attraverso i livelli, soprattutto dal database fino all'interfaccia grafica, abbiamo fatto uso di una classe DTO (Data Transfer Object), che ha consentito di disaccoppiare i layer e mantenere un'architettura pulita e coerente, e delle classi di tipo Gestione che hanno consentito di separare la logica implementativa da quella dell'interfaccia.

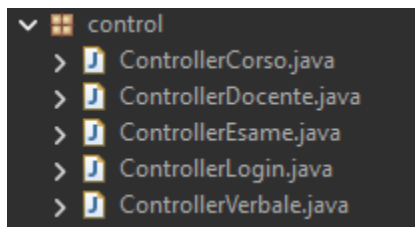
### 5.1 Package Database



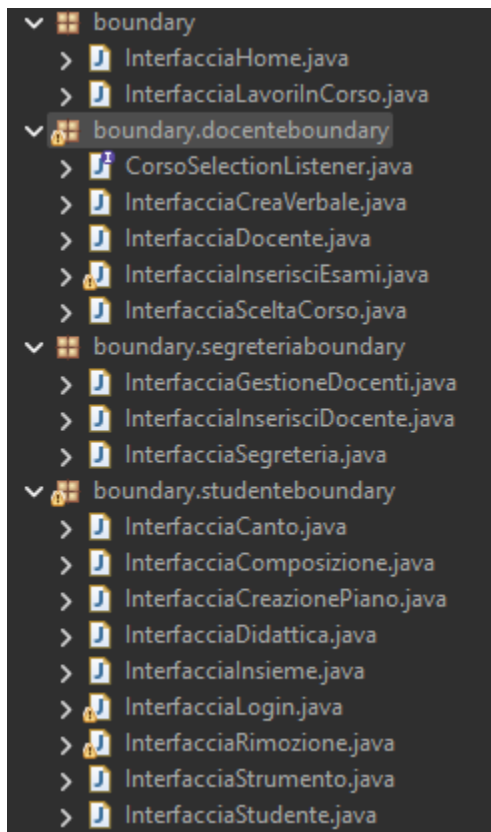
### 5.2 Package Entity



## 5.3 Package Controller

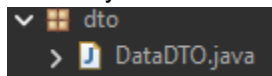


## 5.4 Package Boundary



## 5.5 Package DTO

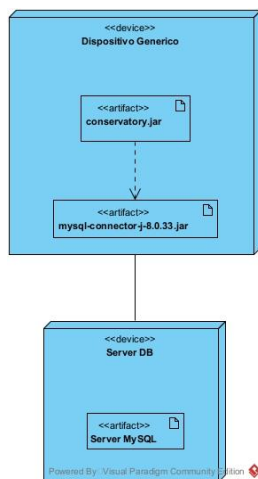
L'introduzione di tale package, estraneo al pattern BCED, nasce dall'esigenza di mostrare sulla GUI collezioni di elementi. Da questo punto di vista, il problema principale è proprio quello che, qualora una determinata chiamata a funzione restituisse alla GUI un elenco di entity, questa, per poterlo 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.



## 5.6 Diagramma di deployment

È necessario disporre di una Java Runtime Environment (JRE) in versione 24 o superiore, cioè l'ambiente necessario per eseguire programmi Java sul computer. Inoltre, affinché il software funzioni correttamente è richiesta la presenza della libreria mysql-connector-j-8.0.33.jar, indispensabile per consentire al package responsabile della gestione del database di interagire con il DBMS relazionale MySQL. È inoltre necessario che sia attiva una connessione a un server MySQL.

La distribuzione del software prevede un'unica versione.





## 6. Testing

### 6.1 Test strutturale

Per eseguire un corretto test strutturale, è importante capire qual è la complessità ciclomatica dei metodi da testare: l'analisi della complessità ciclomatica avviene tramite lo studio dei cammini di esecuzione che ci permettono di individuare il numero di nodi e archi, i quali vengono usati nella formula:

$$V(G) = E - N + 2P$$

Dove:

- **E** = numero di archi (edges)
- **N** = numero di nodi
- **P** = numero di componenti connesse

### 6.1.1 Autenticazione

```
public boolean autenticazione(String username, String password) throws LoginFailedException {
    System.out.println(x:"GestioneStudenti: Avvio processo di autenticazione.");

    Studente foundStudent = stDAO.getCredenziali(username);

    if (foundStudent == null) {
        System.out.println("GestioneStudenti: Studente con username '" + username + "' non trovato nel database.");
        throw new LoginFailedException(message:"Username non esistente.");
    }

    boolean isAuthenticated = foundStudent.checkPassword(password);

    if (isAuthenticated) {
        System.out.println("GestioneStudenti: Autenticazione riuscita per username: " + username);
        return true;
    } else {
        System.out.println("GestioneStudenti: Autenticazione fallita per username: " + username + " (password errata).");
        throw new LoginFailedException(message:"Password non corretta.");
    }
}
```

#### NUMERO CICLOMATICO:

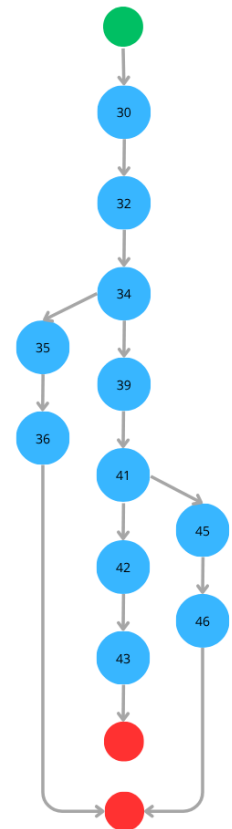
- numero di regioni chiuse del grafo:  $2 + 1 = 3$
- numero di nodi predicati (34, 41) + 1 = 3
- $\# \text{ archi} - \# \text{ nodi} + 2 = (12 - 11) + 2 = 3$

#### CAMMINI INDIPENDENTI:

- 1)  $30 \rightarrow 32 \rightarrow 34 \rightarrow 39 \rightarrow 41 \rightarrow 42 \rightarrow 43$
- 2)  $30 \rightarrow 32 \rightarrow 34 \rightarrow 39 \rightarrow 41 \rightarrow 45 \rightarrow 46$
- 3)  $30 \rightarrow 32 \rightarrow 34 \rightarrow 35 \rightarrow 36$

#### CASI DI TEST:

Test Case	Input	Output Atteso	Cammini
TC1	username valido, password corretta	return true	C1
TC2	username valido, password errata	throw LoginFailed Exception	C2
TC3	username non esistente	throw LoginFailed Exception	C3



### 6.1.2 AggiungiDocente

```
private void inserisciDocente() {
    String nome = txtNome.getText().trim();
    String cognome = txtCognome.getText().trim();
    String password = new String(passwordField.getPassword()).trim();

    boolean valido = true;
    StringBuilder messaggiErrore = new StringBuilder();

    if (!checkEmpty(nome, cognome, password)) {
        return;
    }

    if (!checkNome(nome, messaggiErrore)) {
        valido = false;
    }

    if (!checkCognome(cognome, messaggiErrore)) {
        valido = false;
    }

    if (!checkPassword(password, messaggiErrore)) {
        valido = false;
    }

    if (!valido) {
        JOptionPane.showMessageDialog(this, messaggiErrore.toString(), title:"Errore di validazione", JOptionPane.ERROR_MESSAGE);
        return;
    }

    // Se tutti i controlli sono superati, procedi con l'inserimento
    ControllerDocente CC = ControllerDocente.getInstance();
    String esito = CC.aggiungiDocente(nome, cognome, password);
}
```

```
    if (esito == null) {
        JOptionPane.showMessageDialog(this,
            message:"La matricola è già presente nel database.",
            title:"Errore",
            JOptionPane.WARNING_MESSAGE);
    } else {
        JOptionPane.showMessageDialog(this,
            "Inserimento avvenuto con successo.\nDati inseriti: " + esito + ", " + nome + ", " + cognome,
            title:"Successo",
            JOptionPane.INFORMATION_MESSAGE);
    }

    txtNome.setText(t:"");
    txtCognome.setText(t:"");
    passwordField.setText(t:"");
}
}
```

### NUMERO CICLOMATICO:

- numero di regioni chiuse del grafo:  $6 + 1 = 7$
- numero di nodi predicati (if ed else if) =  $6 + 1 = 7$
- $\# \text{ archi} - \# \text{ nodi} + 2 = (25-20) + 2 = 7$

### CAMMINI INDIPENDENTI:

227 → 228 → 229 → 231 → 232 → 234 → 239  
→ 242 → 246 → 250 → 256 → 257 → 259 →  
265

227 → 228 → 229 → 231 → 232 → 234 → 227

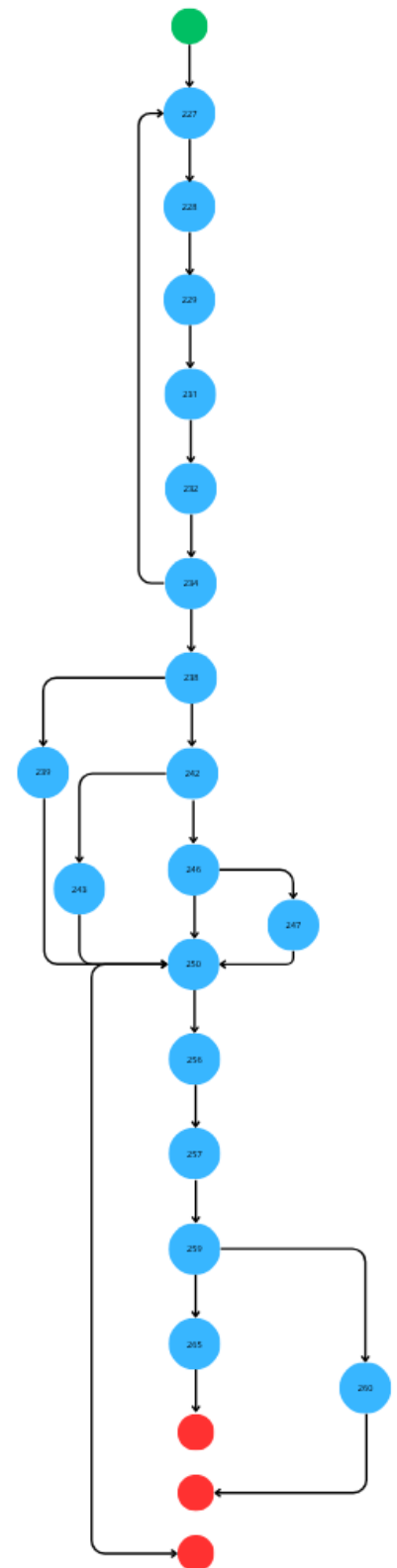
227 → 228 → 229 → 231 → 232 → 234 → 238  
→ 239 → 250 → 256 → 257 → 259 → 265

227 → 228 → 229 → 231 → 232 → 234 →  
→ 238 → 242 → 243 →  
250 → 256 → 257 → 259 → 265

227 → 228 → 229 → 231 → 232 → 234 → 238  
→ 242 → 246 → 247 →  
250 → 256 → 257 → 259 → 265

226 → 227 → 228 → 229 → 231 → 232 → 234  
→ 238 → 239 → 242 → 246 → 267 → 274

227 → 228 → 229 → 231 → 232 → 234 → 238  
→ 242 → 246 → 250 → 256 → 257 → 259 →  
260



### 6.1.3 SalvaPianoDiStudi

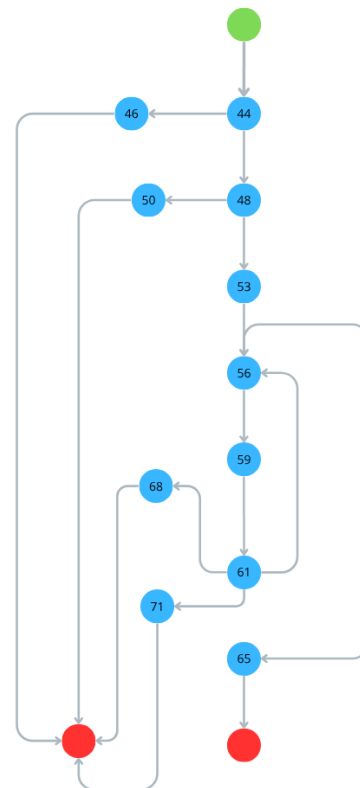
```
public boolean salvaPianoDiStudi(int idpiano, List<CorsoDiStudio> corsiSelezionati) {  
    if (corsiSelezionati == null || corsiSelezionati.isEmpty()) {  
        System.out.println("PianoDiStudiDAO: Nessun corso da salvare.");  
        return false;  
    }  
    if (idpiano <= 0) {  
        System.out.println("PianoDiStudiDAO: ID piano di studi non valido.");  
        return false;  
    }  
    try {  
        for (int i = 0; i < corsiSelezionati.size(); i++) {  
            String insertSql = "INSERT INTO corsiselezionati (idpiano, idcorso) VALUES (" + idpiano + ", " + corsiSelezionati.get(i).getCodice() + ")";  
            DBManager.updateQuery(insertSql);  
        }  
        return true;  
    } catch (SQLException e) {  
        System.err.println("PianoDiStudiDAO: Errore database durante il salvataggio piano studi: " + e.getMessage());  
        return false;  
    } catch (ClassNotFoundException e) {  
        System.err.println("PianoDiStudiDAO: Classe non trovata nel database: " + e.getMessage());  
        return false;  
    }  
}
```

#### NUMERO CICLOMATICO:

- numero di regioni chiuse del grafo:  $5 + 1 = 6$
- numero di nodi predicati (44,48,56,61,61) + 1 = 6
- $\# \text{ archi} - \# \text{ nodi} + 2 = (15 - 11) + 2 = 6$

#### CAMMINI INDIPENDENTI:

- 1)  $44 \rightarrow 46$
- 2)  $44 \rightarrow 48 \rightarrow 50$
- 3)  $44 \rightarrow 48 \rightarrow 53 \rightarrow 56$
- 4)  $44 \rightarrow 48 \rightarrow 53 \rightarrow 56 \rightarrow 59 \rightarrow 61 \rightarrow 56 \rightarrow 65$
- 5)  $44 \rightarrow 48 \rightarrow 53 \rightarrow 56 \rightarrow 59 \rightarrow 61 \rightarrow 68$
- 6)  $44 \rightarrow 48 \rightarrow 53 \rightarrow 56 \rightarrow 59 \rightarrow 61 \rightarrow 71$



### 6.1.4 CreaVerbale

```
private boolean controllaCampi(String data, String matricola) {
    StringBuilder errori = new StringBuilder();

    if (matricola.isEmpty()) {
        errori.append(str:"Il campo matricola è obbligatorio.\n");
        txtMatricola.setText(t:"");
    } else if (!matricola.matches(regex:"D\\d{3}")) {
        errori.append(str:"Formato matricola non valido. Deve essere nel formato D seguito da 3 cifre (es: D123).\n");
        txtMatricola.setText(t:"");
    }

    if (data.isEmpty()) {
        errori.append(str:"Il campo data è obbligatorio.\n");
        txtData.setText(t:"");
    } else if (!private boolean isDataValida(String data)
        boolean boundary.docenteboundary.InterfacciaCreaVerbale.isDataValida(String data)
    ) {
        errori.append(str:"Non possiamo ancora predire il futuro.\n");
        txtData.setText(t:"");
    }

    if (errori.length() > 0) {
        JOptionPane.showMessageDialog(this,
            errori.toString(),
            title:"Errore!",
            JOptionPane.ERROR_MESSAGE);

        return false;
    }

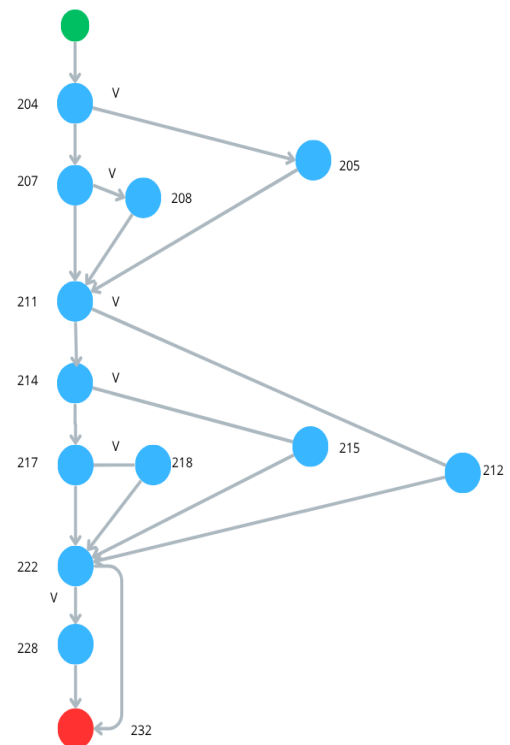
    return true;
}
```

#### NUMERO CICLOMATICO:

n regioni chiuse +1 :  $6 + 1 = 7$

n archi predicato +1 :  $6 + 1 = 7$

#archi - #nodi :  $18 - 15 + 2 = 7$



#### CAMMINI INDIPENDENTI:

- [illegible]