

***Progetto***

**GESTIONE LEZIONI ASSOCIAZIONE  
MUSICALE**



**GLAM**

GESTIONE LEZIONI ASSOCIAZIONE MUSICALE

*Team*

**SEA<sup>2</sup>**



**Studenti:**

**Antonio Pio Montagna  
Salvatore Raiola  
Emmanuel Russo  
Antonio Varese**

# Indice

<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 .....	7
2.3 - Glossario dei termini.....	8
2.4 - Classificazione dei requisiti .....	9
2.4.1 – Requisiti funzionali .....	9
2.4.2 – Requisiti sui dati .....	10
2.4.3 – Vincoli / Altri requisiti .....	10
2.5 – Modellazione dei casi d’uso.....	11
2.5.1 – Attori e casi d’uso .....	11
2.5.2 – Diagramma dei casi d’uso .....	13
2.5.3 – Scenari .....	14
2.6 – Diagramma delle classi.....	18
2.7 – Diagrammi di sequenza.....	20
<b>3 – Stima dei costi .....</b>	<b>24</b>
<b>4 – Piano di test funzionale.....</b>	<b>27</b>
<b>5 – Progettazione .....</b>	<b>46</b>
5.1 – Diagramma delle classi.....	46
5.2 – Diagrammi di sequenza.....	47
<b>6 – Implementazione .....</b>	<b>51</b>
<b>7 - Testing.....</b>	<b>54</b>
7.1 – Test funzionale .....	54

# 1 - Specifiche informali

Si vuole realizzare una applicazione web per la gestione delle prenotazioni delle lezioni in una scuola serale di musica.

L'associazione che gestisce la scuola promuove l'apprendimento della musica a qualunque età, attraverso lezioni da parte di maestri, che offrono la propria disponibilità per due giorni a settimana per lezioni a prezzi modici.

Le lezioni si tengono nella scuola in tutti i giorni della settimana e sono individuali, tutte della durata di 1 h, tra le 17 e le 21. Le lezioni sono di due livelli – principianti e intermedio. Il costo di ogni lezione dipende dal livello (non dal maestro né dallo strumento).

La segreteria della scuola registra le informazioni relative agli insegnanti: nome, il cognome, una foto, gli strumenti (es.: chitarra classica, chitarra jazz, mandolino, pianoforte, etc.), e-mail, numero di telefono. I maestri si collegano al sistema per inserire/modificare i giorni di disponibilità, e per consultare le lezioni svolte mese per mese. Tutti i maestri sono capaci di impartire lezioni sia a principianti, sia di livello intermedio.

Il sistema offre agli aspiranti musicisti la possibilità di cercare i maestri in base allo strumento, di visualizzare per ciascuno di essi i giorni di disponibilità, e di prenotare una o più lezioni.

Gli studenti registrati possono prenotare le lezioni online, effettuando il login tramite username e password. Per la registrazione, lo studente deve fornire alla segreteria cognome, nome, data di nascita, numero di telefono, e-mail. Quando la segreteria della scuola inserisce uno studente, il sistema genera le credenziali, che vengono inviate per e-mail allo studente.

Per effettuare una prenotazione, lo studente sceglie il maestro, la data ed un orario disponibile. Il sistema automaticamente rimuoverà la disponibilità per la fascia oraria prenotata.

La segreteria può visualizzare in qualunque momento le prenotazioni, sia quelle svolte, sia quelle dei giorni a venire.

Se necessario, un discente deve poter disdire una lezione prenotata: in caso di disdetta, il sistema invia una notifica per e-mail alla segreteria. Se la disdetta avviene con almeno 3 giorni d'anticipo, la lezione non deve essere addebitata al discente.

Gli studenti registrati possono visualizzare le date delle lezioni che hanno svolto nel mese, sì da poter effettuare a fine mese il bonifico bancario alla scuola. L'importo da bonificare viene calcolato dal sistema.

Dopo aver consultato i bonifici ricevuti, la segreteria può consultare il sistema per registrare i discenti in regola con i pagamenti. Ogni prima settimana del mese il sistema genera un elenco delle ore di lezione svolte da ogni docente nel mese precedente, in modo che la segreteria possa effettuare i bonifici ai maestri per le lezioni effettivamente svolte, dopo aver decurtato una percentuale (impostabile a tantum dalla segreteria) per la gestione della scuola.

## 2 - Analisi e specifica dei requisiti

### 2.1 - Analisi nomi - verbi

Si vuole realizzare una applicazione web per la gestione delle prenotazioni delle lezioni in una scuola serale di musica.

L'associazione che gestisce la scuola promuove l'apprendimento della musica a qualunque età, attraverso lezioni da parte di maestri, che offrono la propria disponibilità per due giorni a settimana per lezioni a prezzi modici.

Le lezioni si tengono nella scuola in tutti i giorni della settimana e sono individuali, tutte della durata di 1 h, tra le 17 e le 21.

Le lezioni sono di due livelli – principianti e intermedio.

Il costo di ogni lezione dipende dal livello (non dal maestro né dallo strumento).

La segreteria della scuola registra le informazioni relative agli insegnanti: nome, il cognome, una foto, gli strumenti (es.: chitarra classica, chitarra jazz, mandolino, pianoforte, etc.), e-mail, numero di telefono.

I maestri si collegano al sistema per inserire/modificare i giorni di disponibilità, e per consultare le lezioni svolte mese per mese.

Tutti i maestri sono capaci di impartire lezioni sia a principianti, sia di livello intermedio.

Il sistema offre agli aspiranti musicisti la possibilità di cercare i maestri in base allo strumento, di visualizzare per ciascuno di essi i giorni di disponibilità, e di prenotare una o più lezioni.

Gli studenti registrati possono prenotare le lezioni online, effettuando il login tramite username e password.

Per la registrazione, lo studente deve fornire alla segreteria cognome, nome, data di nascita, numero di telefono, e-mail.

Quando la segreteria della scuola inserisce uno studente, il sistema genera le credenziali, che vengono inviate per e-mail allo studente.

Per effettuare una prenotazione, lo studente sceglie il maestro, la data ed un orario disponibile.

Il sistema automaticamente rimuoverà la disponibilità per la fascia oraria prenotata.

La segreteria può visualizzare in qualunque momento le prenotazioni, sia quelle svolte, sia quelle dei giorni a venire.

Se necessario, un discente deve poter disdire una lezione prenotata: in caso di disdetta, il sistema invia una notifica per e-mail alla segreteria.

Se la disdetta avviene con almeno tre giorni d'anticipo, la lezione non deve essere addebitata al discente.

Gli studenti registrati possono visualizzare le date delle lezioni che hanno svolto nel mese, sì da poter effettuare a fine mese il bonifico bancario alla scuola.

L'importo da bonificare viene calcolato dal sistema.

Dopo aver consultato i bonifici ricevuti, la segreteria può consultare il sistema per registrare i discenti in regola con i pagamenti.

Ogni prima settimana del mese il sistema genera un elenco delle ore di lezione svolte da ogni docente nel mese precedente, in modo che la segreteria possa effettuare i bonifici ai maestri per le lezioni effettivamente svolte, dopo aver decurtato una percentuale (impostabile una tantum dalla segreteria) per la gestione della scuola.

## **LEGENDA:**

Classe

Attore

Classe-Attore

Funzionalità

Attributo

## 2.2 – Revisione dei requisiti

1. *Le lezioni si tengono nella scuola in tutti i giorni della settimana.*
2. *Le lezioni sono individuali.*
3. *Le lezioni sono tutte della durata di 1h.*
4. *Le lezioni si svolgono tra le ore 17 e le ore 21.*
5. *Le lezioni sono di due livelli – “principianti” e “intermedio”.*
6. *Le lezioni hanno un costo.*
7. *Il costo di ogni lezione dipende dal livello, non dal maestro né dallo strumento.*
8. *Ogni maestro deve offrire la propria disponibilità per due giorni alla settimana.*
9. *Ogni maestro può impartire lezioni sia a principianti, sia di livello intermedio.*
10. *Il sistema deve offrire alla segreteria una funzionalità per registrare le informazioni relative ad un maestro.*
11. *Di ogni maestro si vuole memorizzare nome, cognome, una foto, uno o più strumenti, e-mail e numero di telefono.*
12. *Il sistema deve offrire al maestro una funzionalità per inserire i propri giorni di disponibilità.*
13. *Il sistema deve offrire al maestro una funzionalità per modificare i propri giorni di disponibilità.*
14. *Il sistema deve offrire al maestro una funzionalità per consultare le proprie lezioni svolte mese per mese.*
15. *Il sistema deve offrire allo studente una funzionalità per cercare i maestri in base allo strumento.*
16. *Il sistema deve offrire allo studente una funzionalità per visualizzare i giorni di disponibilità di un maestro, per ciascuno di essi.*
17. *Il sistema deve offrire alla segreteria una funzionalità per registrare le informazioni relative ad uno studente.*
18. *Per gli studenti registrati si vuole memorizzare all'interno del sistema: nome, cognome, data di nascita, e-mail e numero di telefono.*
19. *Dopo una registrazione, il sistema deve generare le credenziali ed inviarle allo studente tramite e-mail.*
20. *Dopo la registrazione, uno studente registrato potrà effettuare il login nel sistema.*
21. *Uno studente registrato può effettuare tutte le operazioni di uno studente non registrato.*
22. *Il sistema deve offrire allo studente registrato, dopo il login, una funzionalità per prenotare una o più lezioni.*
23. *Per effettuare una prenotazione, lo studente deve specificare il maestro, la data ed un orario disponibile.*
24. *Il sistema deve offrire alla segreteria una funzionalità per visualizzare le prenotazioni, sia quelle svolte che per i giorni a venire.*

25. Il sistema deve offrire allo studente registrato, dopo il login, una funzionalità per disdire una lezione prenotata.
26. Dopo una disdetta, il sistema deve inviare una notifica di disdetta alla segreteria tramite e-mail.
27. Il sistema deve offrire allo studente registrato, dopo il login, una funzionalità per visualizzare le date delle lezioni che ha svolto nel mese.
28. Ogni mese, il sistema deve calcolare automaticamente l'importo che lo studente deve bonificare alla scuola.
29. Il sistema deve offrire alla segreteria una funzionalità per registrare gli studenti in regola con i pagamenti.
30. Ogni prima settimana del mese, il sistema deve generare un elenco delle ore di lezioni svolte da ogni docente nel mese precedente.
31. Il sistema deve offrire alla segreteria una funzionalità per impostare la percentuale di decurtazione.

## 2.3 – Glossario dei termini

Termine	Descrizione	Sinonimi
Lezione	Incontro didattico tra un docente ed uno studente.	
Studente	Persona che desidera ricevere lezioni musicali.	Discente
Studente registrato	Studente che ha effettuato la registrazione.	
Maestro	Persona responsabile dell'insegnamento.	Insegnante, Docente
Segreteria	Ente responsabile dell'amministrazione della scuola.	
Prenotazione	Procedura per riservare in anticipo una lezione.	



## 2.4 – Classificazione dei requisiti

### 2.4.1 – Requisiti funzionali

ID	Requisito	Origine (n. frase dei requisiti revisionati)
RF01	Il sistema deve offrire alla segreteria una funzionalità per registrare le informazioni relative ad un maestro.	10
RF02	Il sistema deve offrire al maestro una funzionalità per inserire i propri giorni di disponibilità.	12
RF03	Il sistema deve offrire al maestro una funzionalità per modificare i propri giorni di disponibilità.	13
RF04	Il sistema deve offrire al maestro una funzionalità per consultare le proprie lezioni svolte mese per mese.	14
RF05	Il sistema deve offrire allo studente una funzionalità per cercare i maestri in base allo strumento.	15
RF06	Il sistema deve offrire allo studente una funzionalità per visualizzare i giorni di disponibilità di un maestro, per ciascuno di essi.	16
RF07	Il sistema deve offrire alla segreteria una funzionalità per registrare le informazioni relative ad uno studente.	17
RF08	Dopo una registrazione, il sistema deve generare le credenziali ed inviarle allo studente tramite e-mail.	19
RF09	Dopo la registrazione, uno studente registrato potrà effettuare il login nel sistema.	20
RF10	Uno studente registrato può effettuare tutte le operazioni di uno studente non registrato.	21
RF11	Il sistema deve offrire allo studente registrato, dopo il login, una funzionalità per prenotare una o più lezioni.	22
RF12	Il sistema deve offrire alla segreteria una funzionalità per visualizzare le prenotazioni, sia quelle svolte che per i giorni a venire.	24
RF13	Il sistema deve offrire allo studente registrato, dopo il login, una funzionalità per disdire una lezione prenotata.	25
RF14	Dopo una disdetta, il sistema deve inviare una notifica di disdetta alla segreteria tramite e-mail.	26
RF15	Il sistema deve offrire allo studente registrato, dopo il login, una funzionalità per visualizzare le date delle lezioni che ha svolto nel mese.	27
RF16	Ogni mese, il sistema deve calcolare automaticamente l'importo che lo studente deve bonificare alla scuola.	28
RF17	Il sistema deve offrire alla segreteria una funzionalità per registrare gli studenti in regola con i pagamenti.	29
RF18	Ogni prima settimana del mese, il sistema deve generare un elenco delle ore di lezioni svolte da ogni docente nel mese precedente.	30

RF19	Il sistema deve offrire alla segreteria una funzionalità per impostare la percentuale di decurtazione.	31
------	--	----

## 2.4.2 – Requisiti sui dati

ID	Requisito	Origine frase requisiti revisionati (n. dei)
RD01	Le lezioni si tengono in tutti i giorni della settimana.	1
RD02	Le lezioni sono individuali.	2
RD03	Le lezioni sono tutte della durata di 1h.	3
RD04	Le lezioni si svolgono tra le ore 17 e le ore 21.	4
RD05	Le lezioni sono di due livelli – “principianti” e “intermedio”.	5
RD06	Le lezioni hanno un costo.	6
RD07	Il costo di ogni lezione dipende dal livello, non dal maestro né dallo strumento.	7
RD08	Ogni maestro deve offrire la propria disponibilità per due giorni alla settimana.	8
RD09	Ogni maestro può impartire lezioni sia a principianti, sia di livello intermedio.	9
RD10	Di ogni maestro si vuole memorizzare nome, cognome, una foto, uno o più strumenti, e-mail e numero di telefono.	11
RD11	Per gli studenti registrati si vuole memorizzare all’interno del sistema: nome, cognome, data di nascita, e-mail e numero di telefono.	18
RD12	Per effettuare una prenotazione, lo studente deve specificare il maestro, la data ed un orario disponibile.	23

## 2.4.3 – Vincoli / Altri requisiti

RNF01 - Per l’invio delle credenziali deve essere disponibile un server di posta elettronica esterno al sistema.

RNF02 - Per l’invio della notifica di avvenuta disdetta deve essere disponibile un server di posta elettronica esterno al sistema.

## 2.5 – Modellazione dei casi d'uso

### 2.5.1 – Attori e casi d'uso

#### **Attori Primari:**

- Studente
- StudenteRegistrato
- Maestro
- Segreteria
- Tempo

#### **Attori Secondari:**

- ServizioMail

#### **Casi d'uso:**

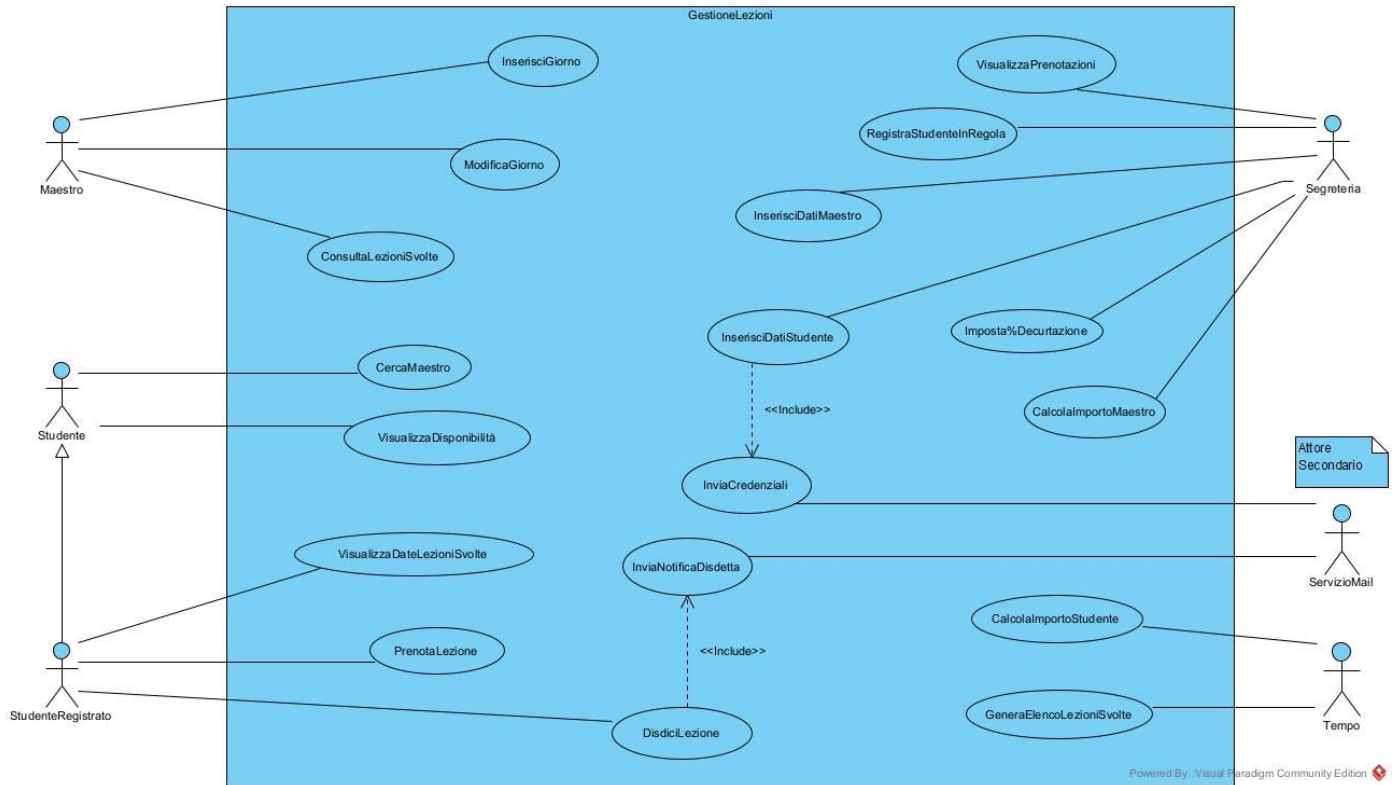
- UC1: CercaMaestro
- UC2: VisualizzaDisponibilità
- UC3: PrenotaLezione
- UC4: DisdiciLezione
- UC5: VisualizzaDateLezioniSvolte
- UC6: InserisciGiorno
- UC7: ModificaGiorno
- UC8: ConsultaLezioniSvolte
- UC9: InserisciDatiMaestro
- UC10: InserisciDatiStudente
- UC11: VisualizzaPrenotazioni
- UC12: Imposta%Decurtazione
- UC13: CalcolaImportoMaestro
- UC14: RegistraStudenteInRegola
- UC15: CalcolaImportoStudente
- UC16: GeneraElencoLezioniSvolte

#### **Casi d'uso d'inclusione:**

- UC17: InviaCredenziali
- UC18: InviaNotificaDisdetta

<b>Caso d'uso</b>	<b>Attori Primari</b>	<b>Attori Secondari</b>	<b>Incl. / Ext.</b>	<b>Requisiti corrispondenti</b>
UC1: CercaMaestro	Studente	-	-	RFo5
UC2: VisualizzaDisponibilità	Studente	-	-	RFo6
UC3: PrenotaLezione	StudenteRegistrato	-	-	RF11
UC4: DisdiciLezione	StudenteRegistrato	-	Include InviaNotificaDisdetta	RF13
UC5: VisualizzaDateLezioniSvolte	StudenteRegistrato	-	-	RF15
UC6: InserisciGiorno	Maestro			RFo2
UC7: ModificaGiorno	Maestro	-	-	RFo3
UC8: ConsultaLezioniSvolte	Maestro	-	-	RFo4
UC9: InserisciDatiMaestro	Segreteria	-	-	RFo1
UC10: InserisciDatiStudente	Segreteria	-	Include InviaCredenziali	RFo7
UC11: VisualizzaPrenotazioni	Segreteria	-	-	RF12
UC12: Imposta%Decurtazione	Segreteria	-	-	RF19
UC13: CalcolaImportoMaestro	Segreteria	-	-	-
UC14: RegistraStudenteInRegola	Segreteria	-	-	RF17
UC15: CalcolaImportoStudente	Tempo	-	-	RF16
UC16: GeneraElencoLezioniSvolte	Tempo	-	-	RF18
UC17: InviaCredenziali	Segreteria	ServizioMail	Incluso in InserisciDatiStudente	RFo8
UC18: InviaNotificaDisdetta	StudenteRegistrato	ServizioMail	Incluso in DisdiciLezione	RF14

## 2.5.2 - Diagramma dei casi d'uso



## 2.5.3 – Scenari

### - UC3: PrenotaLezione

Caso d'uso:	PrenotaLezione
Attore primario	StudenteRegistrato
Attore secondario	-
Descrizione	Uno Studente Registrato, a seguito del login, prenota una o più lezioni per maestro in base allo strumento, in una data, ad un orario e ad un livello di difficoltà.
Pre-Condizioni	Lo studente deve aver effettuato l'accesso e devono esistere una o più lezioni disponibili per la prenotazione.
Sequenza di eventi principale	<ol style="list-style-type: none"><li>1. Il caso d'uso inizia quando lo Studente Registrato richiede la prenotazione della\le lezione\i. <b>1.1</b> Lo Studente Registrato inserisce lo strumento, il maestro, la data, l'orario ed il livello.</li><li>2. Il sistema controlla che la lezione sia disponibile per lo strumento, il maestro, la data, l'ora selezionati. <b>2.1</b> Se la lezione non è disponibile il sistema restituisce un ERRORE allo Studente Registrato.</li><li>3. Il sistema seleziona la\le lezione\i prenotata\le e rimuoverà la\le fascia\le oraria prenotata\le come disponibili.</li></ol>
Post-Condizioni	Lo studente ha prenotato una o più lezioni.
Casi d'uso correlati	-
Sequenza di eventi alternativi	-

## - UC4: DisdiciLezione

<b>Caso d'uso:</b>	<b>Disdici Lezione</b>
<b>Attore primario</b>	StudenteRegistrato
<b>Attore secondario</b>	ServizioMail
<b>Descrizione</b>	Uno studente disdice una lezione prenotata. Una volta disdetta la lezione, il sistema invia una notifica via e-mail alla segreteria per informarla della disdetta.
<b>Pre-Condizioni</b>	Lo studente ha prenotato almeno una lezione.
<b>Sequenza di eventi principale</b>	<ol style="list-style-type: none"> <li>1. Lo studente effettua il login.</li> <li>2. Lo studente visualizza la lista delle lezioni prenotate.</li> <li>3. Lo studente individua la lezione che desidera disdire.</li> <li>4. Lo studente seleziona l'opzione per disdire.</li> <li>5. Lo studente conferma la disdetta.</li> <li>6. Il sistema non addebita il costo se la disdetta è avvenuta con tre giorni di anticipo.</li> <li>7. Il sistema registra la disdetta e rimuove la prenotazione dal sistema.</li> <li>8. Il sistema invia una notifica via e-mail alla segreteria, informandola della disdetta della lezione da parte dello studente.</li> </ol>
<b>Post-Condizioni</b>	La lezione prenotata è stata disdetta correttamente.
<b>Casi d'uso correlati</b>	<i>InviaNotificaDisdetta</i>
<b>Sequenza di eventi alternativi</b>	<ul style="list-style-type: none"> <li>- Se lo studente cambia idea prima di confermare la disdetta, può annullare l'operazione e tornare alla visualizzazione delle prenotazioni.</li> </ul>

## - UC6: InserisciGiorno

<b>Caso d'uso:</b>	<b>InserisciGiorno</b>
<b>Attore primario</b>	Maestro
<b>Attore secondario</b>	-
<b>Descrizione</b>	In ogni momento il Maestro può, attraverso il sistema, inserire giorni di disponibilità in cui potrà tenere lezioni.
<b>Pre-Condizioni</b>	La Data inserita non deve coincidere né con quella in cui viene effettuata l'operazione né con quella di un altro giorno di disponibilità già presente nel sistema.
<b>Sequenza di eventi principale</b>	<ol style="list-style-type: none"> <li>1. Il Maestro inserisce la data del giorno di disponibilità.</li> <li>2. Il sistema verifica la validità dell'INPUT (pre-condition).</li> <li>3. Il sistema crea un nuovo Giorno di disponibilità.</li> <li>4. Il sistema chiede la conferma per l'inserimento.</li> <li>5. Il maestro CONFERMA l'operazione.</li> <li>6. Il nuovo Giorno di disponibilità viene inserito nella lista del sistema.</li> <li>7. Il sistema chiede se si vuole continuare ad inserire.</li> </ol>
<b>Post-Condizioni</b>	Il Giorno di disponibilità indicato dal maestro verrà inserito nel sistema.
<b>Casi d'uso correlati</b>	-
<b>Sequenza di eventi alternativi</b>	<ol style="list-style-type: none"> <li>1. Il Maestro inserisce la data del giorno di disponibilità.</li> <li>2. Il sistema verifica la validità dell'INPUT (pre-condition).</li> <li>3. Il sistema crea un nuovo Giorno di disponibilità.</li> <li>4. Il sistema chiede la conferma per l'inserimento.</li> <li>5. Il maestro NON CONFERMA l'operazione.</li> <li>6. Il nuovo Giorno di disponibilità viene distrutto.</li> <li>7. Il sistema chiede se si vuole continuare ad inserire.</li> </ol>

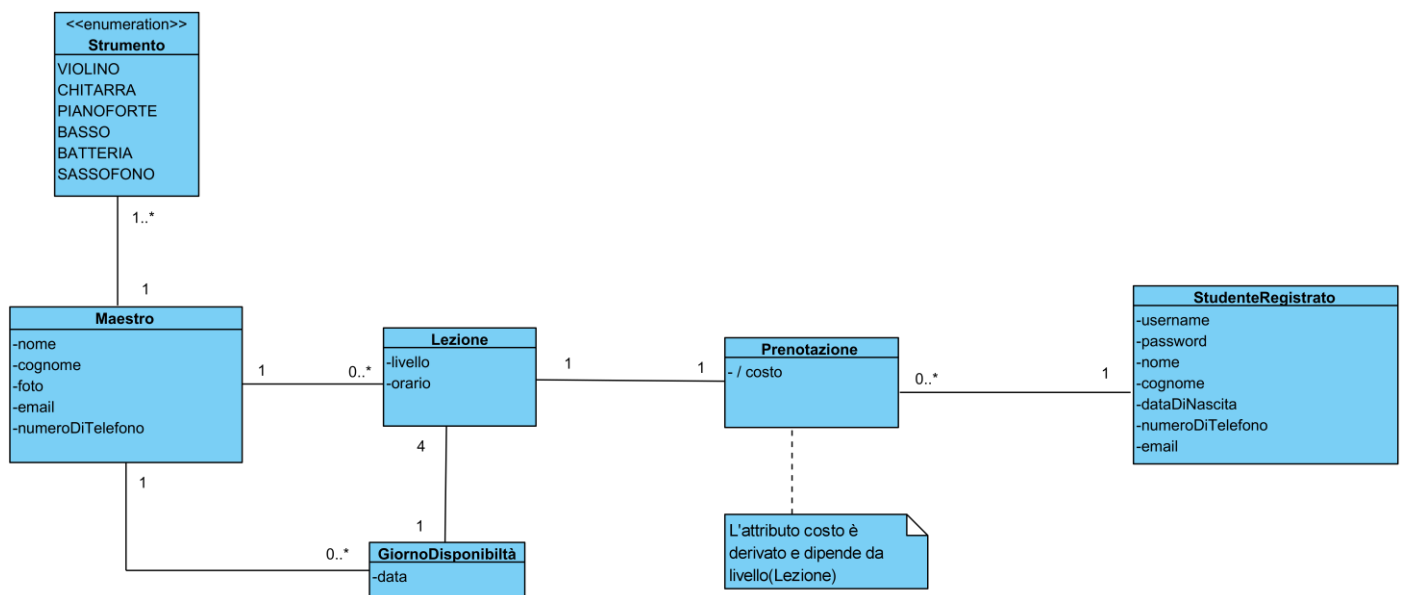


## - UC7: ModificaGiorno

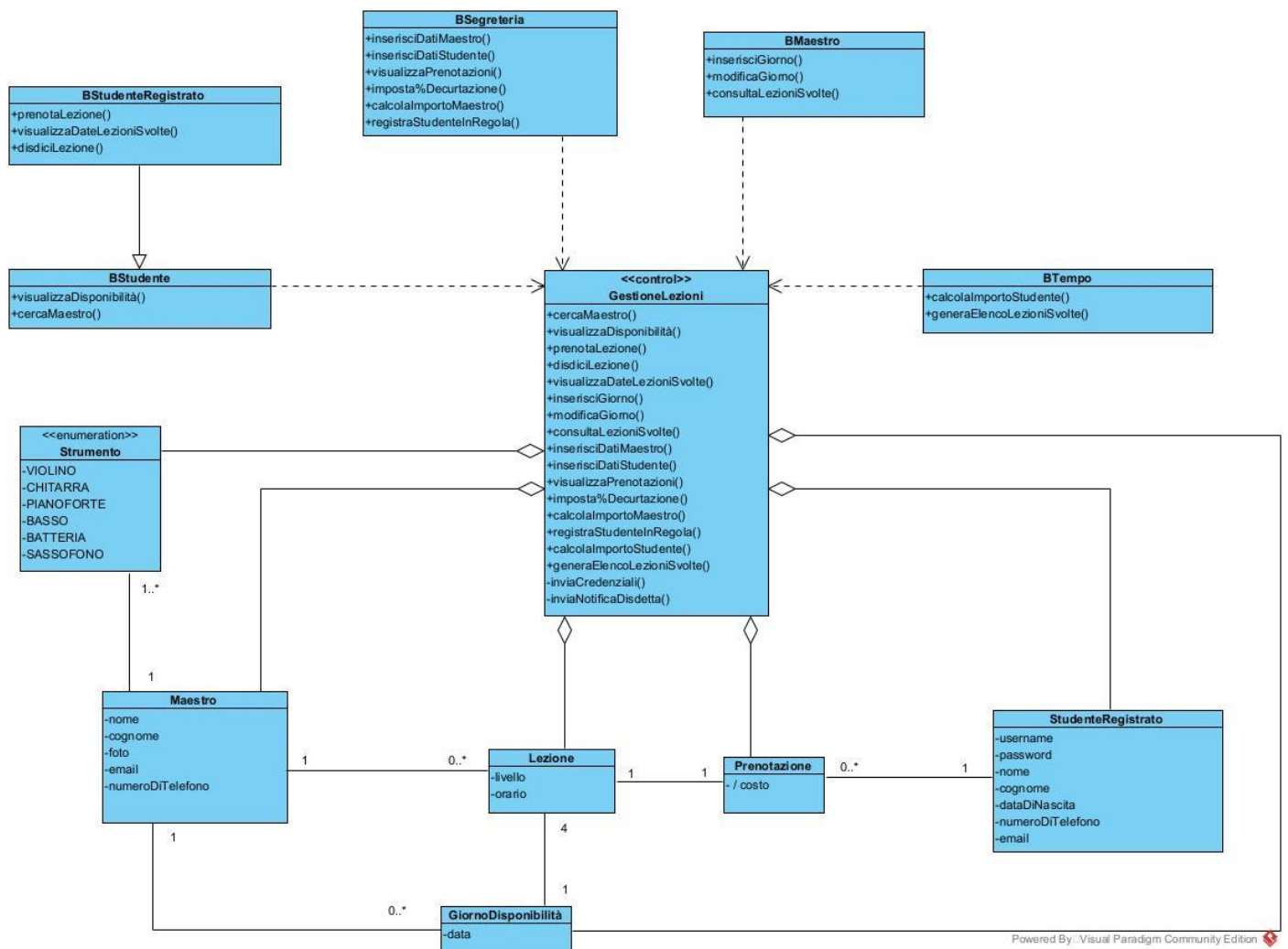
<b>Caso d'uso:</b>	<b>Invio newsletter</b>
<b>Attore primario</b>	Maestro
<b>Attore secondario</b>	-
<b>Descrizione</b>	In ogni momento il Maestro può, attraverso il sistema, modificare una delle date in cui è disponibile per le lezioni.
<b>Pre-Condizioni</b>	Il Maestro ha precedentemente inserito almeno un giorno di disponibilità.
<b>Sequenza di eventi principale</b>	<ol style="list-style-type: none"> <li>1. Il caso d'uso inizia quando il Maestro richiede la modifica di un giorno di disponibilità.</li> <li>2. Il Maestro inserisce la data del giorno di disponibilità che desidera modificare.</li> <li>3. Il sistema verifica la validità di questa data, ovvero se essa è presente tra i giorni di disponibilità precedentemente inseriti dal Maestro.</li> <li>4. Il Maestro inserisce la nuova data in cui è disponibile.</li> <li>5. Il sistema verifica la validità di questa data, ovvero se essa non precede la data di utilizzo del caso d'uso oppure se essa non coincide con la data di un altro giorno di disponibilità esistente.</li> <li>6. Il Maestro conferma di voler effettuare la modifica.</li> <li>7. Il sistema sostituisce la data da modificare con la nuova data di disponibilità.</li> </ol>
<b>Post-Condizioni</b>	La data in cui il Maestro è disponibile per le lezioni è stata modificata correttamente.
<b>Casi d'uso correlati</b>	-
<b>Sequenza di eventi alternativi</b>	<p>Se il Maestro inserisce come data da modificare una data non valida (non presente nei giorni di disponibilità precedentemente inseriti dal Maestro), allora il sistema restituisce messaggio d'errore e termina il caso d'uso.</p> <p>Dopo aver inserito una data da modificare valida, se il Maestro inserisce come nuova data di disponibilità una data precedente alla data d'utilizzo del caso d'uso oppure uguale alla data di un altro giorno di disponibilità esistente, allora il sistema restituisce messaggio d'errore e termina il caso d'uso.</p> <p>Se il Maestro non conferma la modifica oppure inserisce come nuova data di disponibilità una data uguale alla data da modificare, allora il sistema non modifica alcun giorno di disponibilità ed il caso d'uso si conclude.</p>

## 2.6 – Diagramma delle classi

### - Diagramma delle classi di prima analisi

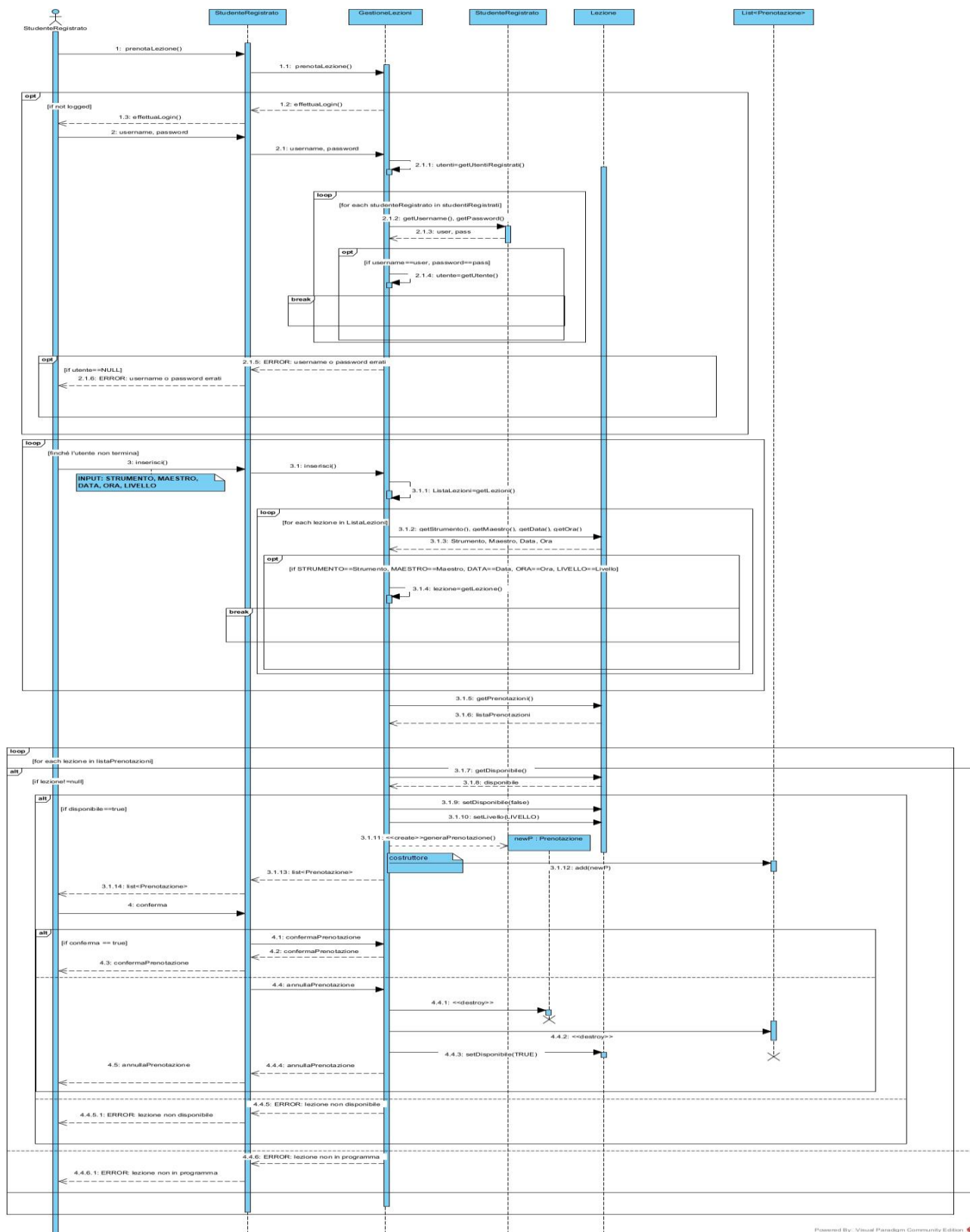


- Diagramma delle classi raffinato secondo il modello BCE



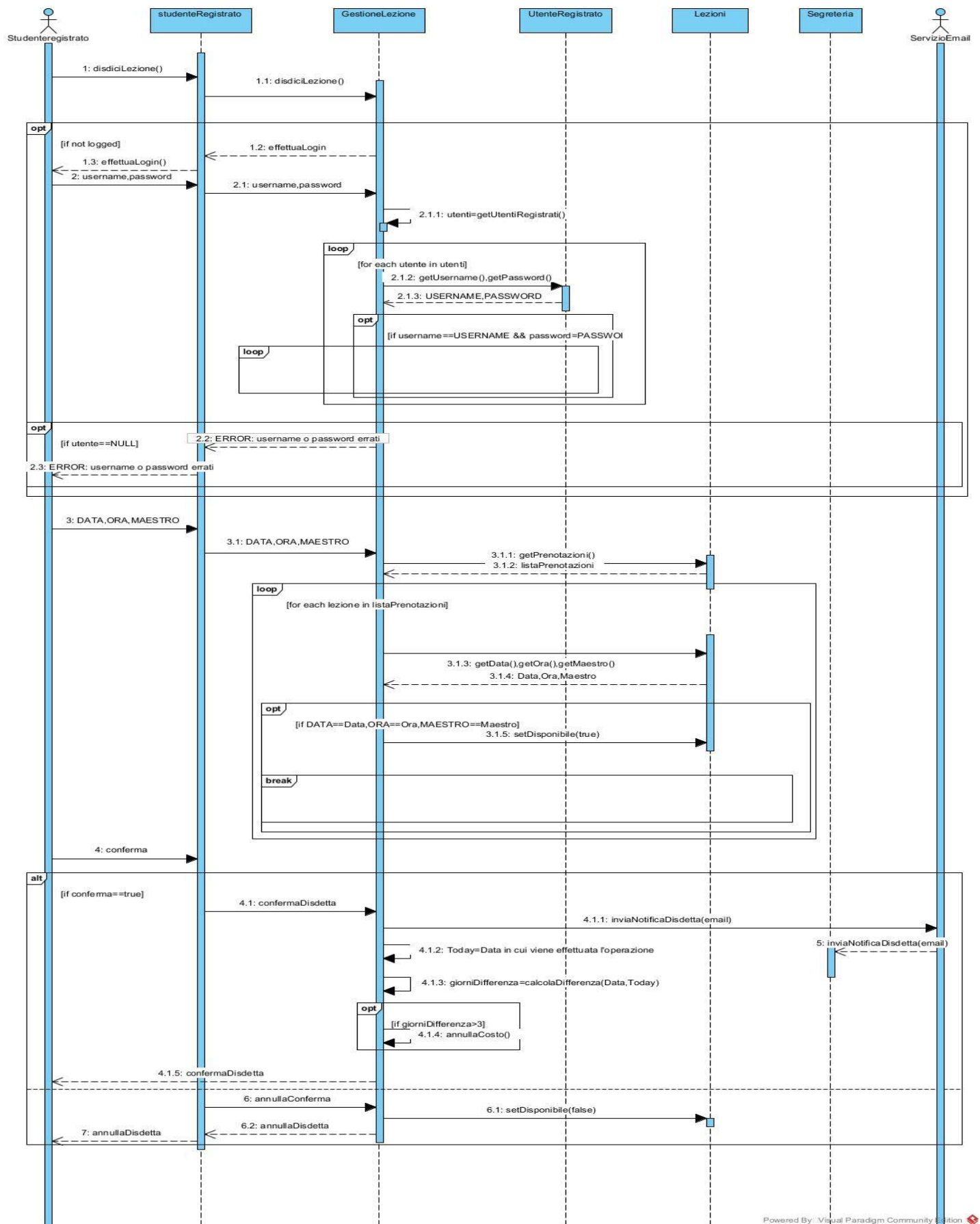
## 2.7 – Diagrammi di sequenza

### UC3: PrenotaLezione



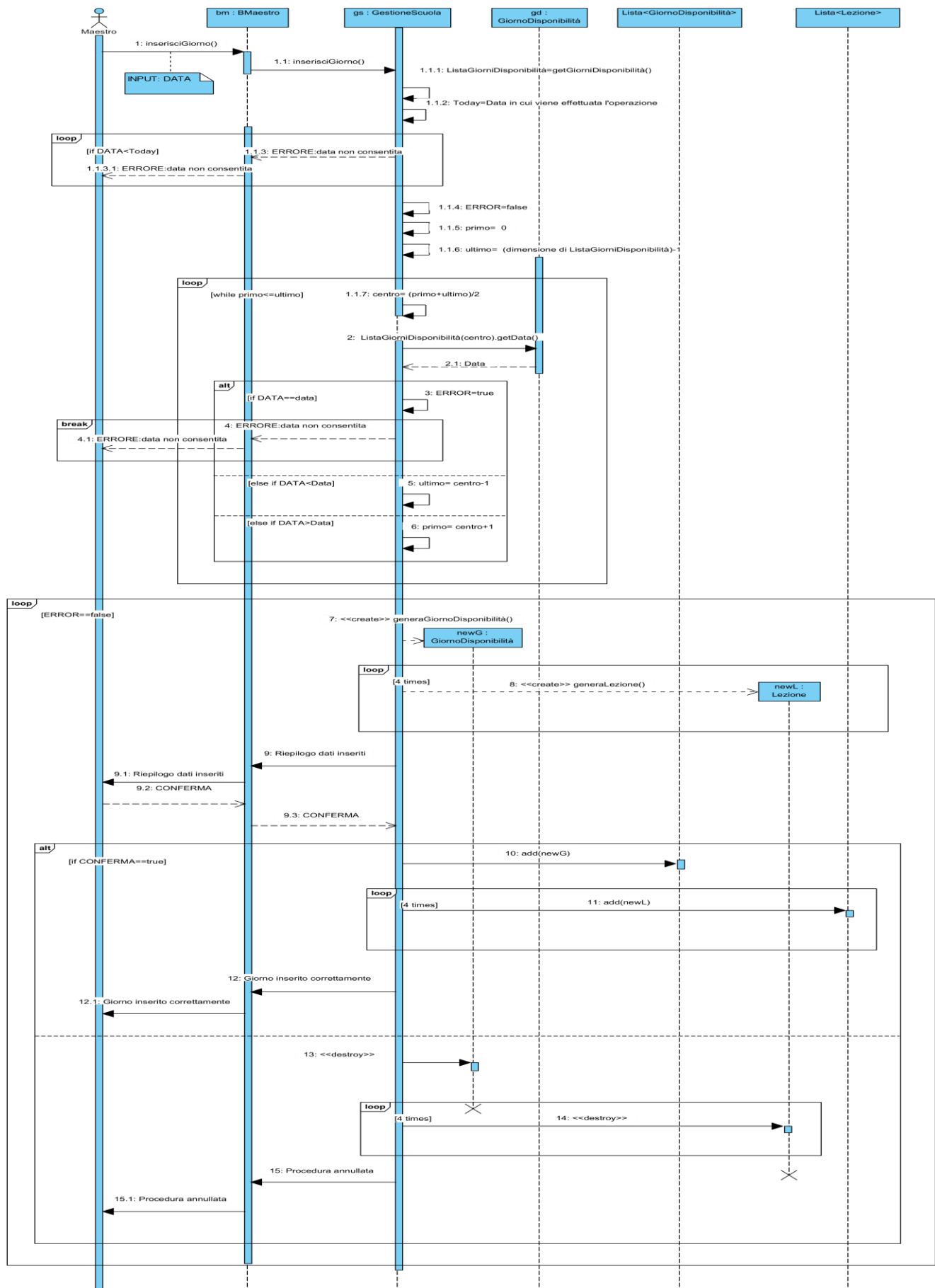
Powered By: Visual Paradigm Community Edition

## UC4: DisdiciLezione

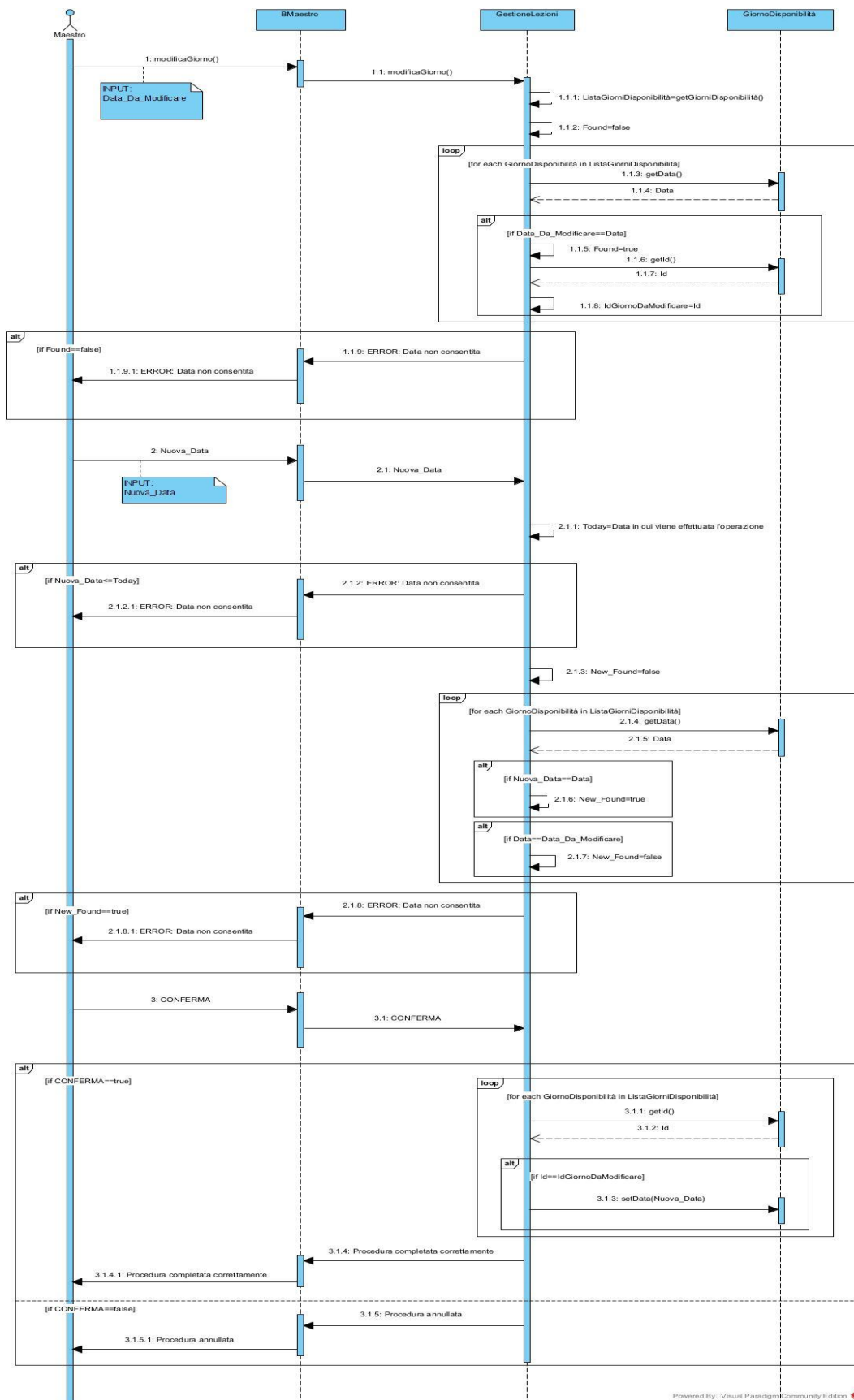


Powered By: Visual Paradigm Community Edition

## UC6: InserisciGiorno



# UC7: ModificaGiorno



Powered By: Visual Paradigm Community Edition

### 3 – Stima dei costi

- Tabella di riferimento per le complessità di dati e transazioni

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

#### - UC3: PrenotaLezione

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
NILF	1		4		4
NEIF	0				
NEI	5	3			15
NEO	1	7			7
NEQ	0				

UFP = 26
LLOC/FP = 1378

$FP = 26 * (0.65 + 0.01 * 32) = 25.22$
JAVA = 1336.66



## - UC4: DisdiciLezione

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
NILF	1		4		4
NEIF	0				
NEI	3	3			9
NEO	1	7			7
NEQ					

UFP = 20
LLOC/FP = 1060

$FP = 20 \cdot (0.65 + 0.01 \cdot 32) = 19.4$
JAVA = 1028.2

## - UC6: InserisciGiorno

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
NILF	1		4		4
NEIF	0				
NEI	1	3			3
NEO	1	7			7
NEQ	0				

UFP = 14
LLOC/FP = 742

FP = 13.58
JAVA = 719.74

## - UC7: ModificaGiorno

	VALORE	SEMPLICE	MEDIO	COMPLESSO	TOT
NILF	1		4		4
NEIF					
NEI	2	3			6
NEO	1	7			7
NEQ					

UFP = 17
LLOC/FP = 901

FP = 16.49
JAVA= 873.97

- Tabella elenco dei fattori correttivi (il cui valore è compreso tra 0 e 5)

### FATTORI CORRETTIVI

COMUNICAZIONE DATI	2
DISTRIBUZIONE ELABORAZIONE	1
PRESTAZIONI	3
UTILIZZO INTENSIVO CONFIGURAZIONE	3
FREQUENZA DELLE TRANSAZIONI	3
INSERIMENTO DATI INTERATTIVO	2
EFFICIENZA PER L'UTENTE FINALE	4
AGGIORNAMENTO INTERATTIVO	3
COMPLESSITA' ELABORATIVA	0
RIUSABILITA'	3
FACILITA' INSTALLAZIONE	2
FACILITA' GESTIONE OPERATIVA	2
MOLTEPLICITA' DI SITI	1
FACILITA' DI MODIFICA	3

## 4 – Piano di test funzionale

### PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* PER LA FUNZIONALITÀ “Prenota Lezione”

Username	Password	Maestro	Data	Ora	Strumento	Livello	Elemento di sistema database
<ul style="list-style-type: none"><li>• Stringa di caratteri di lunghezza &lt;= 50</li><li>• Stringa di caratteri di lunghezza &gt; 50 [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• Stringa di caratteri di lunghezza &gt;=8 e &lt;= 50</li><li>• Stringa di caratteri di lunghezza &gt; 50 [ERROR]</li><li>• Stringa di caratteri di lunghezza &lt;8 [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• Nome stringa caratteri &lt;=50 e Cognome stringa caratteri &lt;=50</li><li>• Nome stringa caratteri &gt;50 [ERROR]</li><li>• Cognome Stringa caratteri &gt;50 [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• Data con formato valido (gg/mm/aaaa)</li><li>• Data con formato non valido [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• Ora con formato valido [hh:mm], compresa tra le 17.00 e le 20.00.</li><li>• Ora con formato non valido [ERROR]</li><li>• Ora non compresa tra le 17.00 e le 20.00 [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• Lo strumento deve essere uno tra i seguenti: violino, chitarra, pianoforte, batteria, basso, sassofono.</li><li>• Altri strumenti non validi [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• Livello della lezione: “principiante” o “intermedio”.</li><li>• Altri livelli non validi [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• Lezione disponibile.</li><li>• Lezione non presente nel database [ERROR]</li><li>• Lezione già prenotata [ERROR]</li><li>• Username e/o password non presenti nel Database [ERROR]</li></ul>

Il numero di test da effettuarsi senza particolari vincoli è:  $2*3*3*2*3*2*2*4 = 1728$

Introduciamo i vincoli [ERROR].

Il numero di test da eseguire per testare singolarmente i vincoli è 13 (1 per Username, 2 per Password, 2 per Maestro, 1 per Data, 2 per Ora, 1 per Strumento, 1 per Livello, 3 per ElementoDiSistemaDatabase).

Il numero di test risultante è  $(1*1*1*1*1*1*1*1) + 13 = 14$

## TEST SUITE

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Ottenuti	Post-condizioni Attese
1	Tutti gli input sono validi.	Username valido,  Password valida,  Maestro valido,  Data valida,  Ora valida,  Strumento valido,  Livello valido,  Elementi di sistema database validi.	Il database è inizializzato correttamente.  Lo studente effettua il login correttamente.  Le lezioni sono disponibili per lo strumento nella data e ora selezionate.	{ Username: "anto24",  Password: "ciaoraga20",  Nome Maestro: "Andrea ",  Cognome Maestro: "Pirlo",  Data: "24/06/2024",  Ora: "18:00",  Strumento: "Sassofono",  Livello: "Base" }	Prenotazione avvenuta con successo.	Lo studente ottiene la sua prenotazione.

2	Username: stringa > 50 caratteri.	Username: stringa > 50 caratteri [ERROR],  Validi:  Password, Maestro, Data, Ora, Strumento, Livello, ElementoDiSiste maDatabase.	Lo studente vuole accedere per effettuare la prenotazione.	{ Username: "aaaaaaaaaaaaa aaaaaaaaaaaaa aaaaaaaaaaaaa aaaaaaaaaaaaa aaaaaaaaaaaaa",  Password: "nocappo813",  Nome Maestro: "Andrea ",  Cognome Maestro: "Pirlo",  Data: "24/06/2024",  Ora: "18:00",  Strumento: "Sassofono",  Livello: "Base" }	Username non valido.	Il sistema chiede nuovamente di inserire l'username.
3	Password: stringa di caratteri di lunghezza > 50	Password: stringa di caratteri di lunghezza >50 [ERROR],  Validi:  Username, Maestro, Data, Ora,	Lo studente registrato seleziona la funzionalità "Prenota Lezione".  Il sistema richiede all'utente registrato	{ Username: "dottoraro2",  Password: "htppp111",  Nome Maestro: "Giancarlo",  Cognome Maestro: "Zita",	Password non valida.	Il sistema chiede nuovamente di inserire la password dell'utente.

		Strumento, Livello, ElementiDiSistemaDatabase.	l'inserimento della password.	Data: "25/06/2024",  Ora: "17:00",  Strumento: "Batteria",  Livello: "Base" }		
4	Password: stringa di caratteri di lunghezza <8	Password: stringa di caratteri di lunghezza <=8 [ERROR],  Validi:  Username, Maestro, Data, Ora, Strumento, Livello, ElementiDiSistemaDatabase.	Lo studente registrato seleziona la funzionalità prenotazione.  Il sistema richiede all'utente registrato l'inserimento della password.	{ Username: "angy7537",  Password: "FNS!1926",  Nome maestro: "Mauro ",  Cognome maestro: "Sassi",  Data: "29/06/2024",  Ora: "20:00",  Strumento: "sassofono",  Livello: "Base" }	Password non valida.	Il sistema chiede nuovamente di inserire la password dell'utente.
5	Nome Maestro: stringa > 50 caratteri.	Nome Maestro stringa > 50 caratteri [ERROR],	Lo studente vuole accedere per effettuare la prenotazione.	{ Username: "aaaaaaaaaaaa", Password: "nocappo813",	Nome maestro non valido.	Il sistema chiede nuovamente di inserire il maestro.

		Validi:  Username, Password, Data, Ora, Strumento, Livello, ElementoDiSistemaDatabase.		Nome maestro: "Andreaaaaaaaaaa aaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaa aaaaaaaaaaaaa",  Cognome Maestro: "Galimberti",  Data: "24/06/2024",  Ora: "18:00",  Strumento: "Sassofono",  Livello: "Base" }		
6	Cognome Maestro: stringa > 50 caratteri.	Cognome Maestro stringa > 50 caratteri [ERROR],  Validi:  Username, Password, Data, Ora, Strumento, Livello, ElementoDiSistemaDatabase.	Lo studente vuole accedere per effettuare la prenotazione.	{ Username: "aaaaaaaaaaaaa", Password: "nocappo813",  Nome Maestro: "Andrea",  Cognome Maestro: "Galimbertiiiiiiii iiiiiiiiiiiiiiiiiiii iiiiiiiiiiiiiiiiiiii iiiiiiiiiiiiiiiiiiii iiiiiiiiiiiiii",  Data: "24/06/2024",  Ora: "18:00",	Cognome maestro non valido.	Il sistema chiede nuovamente di inserire il cognome maestro.

				Strumento: "Sassofono",  Livello: "Base" }		
7	Data con formato non valido.	Data con formato non valido [ERROR],  Validi:  Username, Password, Maestro, Ora, Strumento, Livello, ElementoDiSistemaDatabase.	Lo studente vuole effettuare la prenotazione.  Il sistema richiede all'utente l'inserimento della data.	{ Username: "oaksakoaos",  Password: "nocappo813",  Nome Maestro: "Mario",  Cognome Maestro: "Rossi",  Data: "24/06/24",  Ora: "14:00",  Strumento: "Sassofono",  Livello: "Base" }	Data inserita non esistente oppure formato errato.  Riprovare...	Il sistema chiede di inserire nuovamente la data.
8	Ora con formato non valido.	Ora con formato non valido [ERROR],  Validi:  Username, Password, Maestro, Data,	Lo studente vuole effettuare la prenotazione.  Il sistema richiede all'utente l'inserimento dell'ora.	{ Username: "oaksakoaos",  Password: "nocappo813",  Nome Maestro: "Mario",	Non ci sono Lezioni per l'orario scelto.  Le Lezioni iniziano solo alle 17:00, alle 18:00, alle 19:00	Il sistema chiede nuovamente di inserire l'ora.



		Strumento, Livello, ElementoDiSistemaDatabase.		Cognome Maestro: "Rossi",  Data: "24/06/2024",  Ora: "19",  Strumento: "Sassofono",  Livello: "Base" }	oppure alle 20:00.  Riprovare...	
9	Ora non compresa tra le 17.00 e le 20.00.	Ora non compresa tra le 17.00 e le 20.00 [ERROR],  Ora con formato valido,  Validi:  Username, Password, Maestro, Data, Strumento, Livello, ElementoDiSistemaDatabase.	Lo studente vuole effettuare la prenotazione.  Il sistema richiede all'utente l'inserimento dell'ora.	{ Username: "oaksakoaos",  Password: "nocappo813",  Nome Maestro: "Mario",  Cognome Maestro: "Rossi",  Data: "24/06/2024",  Ora: "14:00",  Strumento: "Sassofono",  Livello: "Base" }	Non ci sono Lezioni per l'orario scelto.  Le Lezioni iniziano solo alle 17:00, alle 18:00, alle 19:00 oppure alle 20:00.  Riprovare...	Il sistema chiede nuovamente di inserire l'ora.
10	Altri strumenti non validi.	Strumento non valido [ERROR],  Validi:	Lo studente vuole effettuare la prenotazione.	{ Username: "oaksakoaos",  Password:	Non vengono erogate lezioni per lo	Il sistema chiede di inserire nuovamente lo strumento.

		Username, Password, Maestro, Data, Ora, Livello, ElementoDiSistemaDatabase.	Il sistema richiede all'utente l'inserimento dello strumento.	"nocappo813",  Nome Maestro: "Mario",  Cognome Maestro: "Rossi",  Data: "24/06/2024",  Ora: "18:00",  Strumento: "Clarinetto",  Livello: "Base" }	Strumento scelto.  Riprovare scegliendo un altro strumento...	
11	Altri livelli non validi.	Livello non valido [ERROR],  Validi:  Username, Password, Maestro, Data, Ora, Strumento, ElementiDiSistemaDatabase.	Il livello della lezione inserito non è presente nel sistema.	{ Username: "salvo10ra",  Password: "noleg39day",  Nome Maestro: "Ludovico",  Cognome Maestro: "Lemme",  Data: "14/06/2024",  Ora: "18:00",  Strumento: "Chitarra",  Livello: "Avanzato" }	Le lezioni possono essere solo di 2 livelli: BASE (costo di 15.00 euro) oppure INTERMEDIO (costo di 30.00 euro).	Il sistema chiede nuovamente di inserire il livello della lezione.

12	Lezione non presente nel database [ERROR]	<p>ElementoDiSistemaDatabase non valido [ERROR],</p> <p>Validi:</p> <p>Username, Password, Maestro, Data, Ora, Strumento, Livello.</p>	La lezione non risulta nel database.	<p>{</p> <p>Username: "lory446",</p> <p>Password: "circuitodon",</p> <p>Nome Maestro: "Andrea",</p> <p>Cognome Maestro: "Presti",</p> <p>Data: "13/06/2024",</p> <p>Ora: "21:00",</p> <p>Strumento: "Batteria",</p> <p>Livello: "Intermedio"</p> <p>}</p>	Lezione non trovata.	Il sistema non trova lezione in input e non può procedere per la prenotazione.
13	Lezione già prenotata [ERROR]	<p>ElementoDiSistemaDatabase non valido [ERROR],</p> <p>Validi:</p> <p>Username, Password, Maestro, Data, Ora, Strumento, Livello.</p>	La lezione risulta essere già prenotata.	<p>{</p> <p>Username: "pablo72",</p> <p>Password: "aitriplei22",</p> <p>Nome Maestro: "Stefano",</p> <p>Cognome Maestro: "Pioli",</p> <p>Data: "26/06/2024",</p>	La lezione non è disponibile.	<p>Il sistema non può prenotare la lezione.</p> <p>Il sistema richiede di inserire nuovamente i dati della lezione.</p>

				Ora: "19:00",  Strumento: "Pianoforte",  Livello: "Base" }		
14	Username e/o password non presenti nel Database [ERROR]	ElementoDiSistemaDatabase non valido [ERROR],  Validi:  Username, Password, Maestro, Data, Ora, Strumento, Livello.	Username e/o password utente non sono presenti nel Database.	{ Username: "manu223",  Password: "Maestroregistrato",  Nome maestro: "Mario",  Cognome maestro: "Rossi",  Data: "22/07/2024",  Ora: "17:00",  Strumento: "Sassofono",  Livello: "Intermedio" }	Studente non registrato.	Il sistema non può prenotare la lezione.

## PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* PER LA FUNZIONALITÀ “Disdici Lezione”

Nome Maestro	Cognome Maestro	Data	Ora	Elemento di sistema database
<ul style="list-style-type: none"><li>Nome stringa di caratteri di lunghezza <math>\leq 50</math></li><li>Nome stringa di caratteri di lunghezza <math>&gt; 50</math> [ERROR]</li></ul>	<ul style="list-style-type: none"><li>Cognome stringa di caratteri di lunghezza <math>\leq 50</math></li><li>Cognome stringa di caratteri di lunghezza <math>&gt; 50</math> [ERROR]</li></ul>	<ul style="list-style-type: none"><li>Data con formato valido (gg/mm/aaa)</li><li>Data con formato non valido [ERROR]</li></ul>	<ul style="list-style-type: none"><li>Ora con formato valido (hh.mm), compresa tra le 17.00 e le 20.00.</li><li>Ora con formato non valido [ERROR]</li><li>Ora non compresa tra le 17.00 e le 20.00 [ERROR]</li></ul>	<ul style="list-style-type: none"><li>La lezione è presente nel database.</li><li>Lezione non presente nel database [ERROR]</li><li>Lezione non prenotata [ERROR]</li></ul>

Il numero di test da effettuarsi senza particolari vincoli è:  $2*2*2*3*3 = 72$

Introduciamo i vincoli [ERROR].

Il numero di test da eseguire per testare singolarmente i vincoli è 7 (1 per Nome Maestro, 1 per Cognome Maestro, 1 per Data, 2 per Ora, 2 per Elemento di sistema database).

Il numero di test risultante è  $(1*1*1*1*1) + 7 = 8$

## TEST SUITE

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Ottenuti	Post-condizioni Attese
1	Tutti gli input sono validi.	Nome Maestro valido,  Cognome Maestro valido,  Data valida,  Ora valida,  ElementoDi SistemaData base valido.	Lo studente ha effettuato il login.  Il database è inizializzato correttamente.  La lezione esiste per la data, ora e per il maestro selezionato.	{ Nome Maestro: "Andrea",  Cognome Maestro: "Pirlo",  Data: "27/06/2024",  Ora: "19.00" }	DISDETTA AVVENUTA CON SUCCESSO!	La lezione è stata disdetta correttamente.  Il sistema notifica via mail la disdetta alla segreteria.
2	Nome Maestro: stringa > 50 caratteri	Nome Maestro stringa > 50 caratteri [ERROR],  Validi:  Cognome, Data, Ora, ElementoDi SistemaData base.	Lo studente ha inserito il maestro per la disdetta della lezione.	{ Nome Maestro: "Andreaaaaaaaaaaaaaaaaaaaaaaaaaaaaa",  Cognome Maestro: "Pirlo",  Data: "24/06/2024",  Ora: "18:00" }	Nome maestro non valido.	Il sistema chiede nuovamente di inserire il nome maestro.
3	Cognome Maestro: stringa > 50 caratteri	Cognome Maestro stringa > 50 caratteri [ERROR],	Lo studente ha inserito il maestro per la disdetta della lezione.	{ Nome Maestro: "Andrea",	Cognome maestro non valido	Il sistema chiede nuovamente di inserire il

		Validi:  Nome, Data, Ora, ElementoDi SistemaData base.		Cognome Maestro: "Pirlooooooooooooo oooooooooooooooooooo oooooooooooooooooooo oooooooooooooooooooo",  Data: "24/06/2024",  Ora: "18:00" }		cognome maestro.
4	Data con formato non valido.	Data con formato non valido [ERROR],  Validi:  Nome, Cognome, Ora, ElementoDi SistemaData base.	Lo studente vuole effettuare la disdetta.	{ Nome Maestro: "Mario",  Cognome Maestro: "Rossi",  Data: "24/06/24",  Ora: "18:00" }	Data inserita non esistente oppure formato errato.  Riprovare...	Il sistema chiede nuovamente di inserire la data.
5	Ora con formato non valido.	Ora con formato non valido [ERROR],  Validi:  Nome, Cognome, Data, ElementoDi SistemaData base.	Lo studente vuole effettuare la disdetta.	{ Nome Maestro: "Mario",  Cognome Maestro: "Rossi",  Data: "24/06/2024",  Ora: "18" }	Non ci sono Lezioni per l'orario scelto.  Le Lezioni iniziano solo alle 17:00, alle 18:00, alle 19:00 oppure alle 20:00.  Riprovare...	Il sistema chiede nuovamente di inserire l'ora.

6	Ora non compresa tra le 17.00 e le 20.00.	Ora non compresa tra le 17.00 e le 20.00 [ERROR],  Validi:  Nome, Cognome, Data, ElementoDi SistemaDataBase.	Lo studente vuole effettuare la disdetta.	{ Nome Maestro: "Mario"  Cognome Maestro: "Rossi",  Data: "24/06/2024",  Ora: "14.00" }	Non ci sono Lezioni per l'orario scelto.  Le Lezioni iniziano solo alle 17:00, alle 18:00, alle 19:00 oppure alle 20:00.  Riprovare...	Il sistema chiede nuovamente di inserire l'ora.
7	Lezione non presente nel database [ERROR].	ElementoDi SistemaDataBase non valido [ERROR],  Validi:  Nome, Cognome, Data, Ora.	La lezione non è stata prenotata o è stata già disdetta.	{ Nome Maestro: "Stefano",  Cognome Maestro: "Pioli",  Data: "26/06/2024",  Ora: "19:00" }	Lezione non trovata.	Il sistema non può effettuare la disdetta per la lezione richiesta.  Il sistema richiede di inserire nuovamente i dati della lezione.
8	Lezione non prenotata [ERROR].	ElementoDi SistemaDataBase non valido [ERROR],  Validi:  Maestro, Ora, Data.	La lezione non è stata prenotata.	{ Nome Maestro: "Stefano",  Cognome Maestro: "Pioli",  Data: "26/06/2024",  Ora: "19:00" }	Lezione non trovata.	Il sistema non può effettuare la disdetta per la lezione richiesta.



## PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* PER LA FUNZIONALITÀ “Inserisci Giorno”

Data	Elemento di sistema database
<ul style="list-style-type: none"><li>Data con formato valido (gg/mm/aaaa).</li><li>Data con formato non valido [ERROR]</li></ul>	<ul style="list-style-type: none"><li>Data non presente nel database.</li><li>Data già presente nel database [ERROR]</li></ul>

Il numero di test da effettuarsi senza particolari vincoli è:  $2 \times 2 = 4$

Introduciamo i vincoli [ERROR].

Il numero di test da eseguire per testare singolarmente i vincoli è 2 (1 per Data, 1 per ElementoDiSistemaDatabase).

Il numero di test risultante è  $(1 \times 1) + 2 = 3$

### TEST SUITE

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Tutti gli input sono validi.	Data con formato valido.  Elementi di sistema database validi.	Il database è inizializzato correttamente. Il maestro ha effettuato l'accesso.	{  Data: "09/07/2024" }	Un nuovo giorno di disponibilità è stato inserito con successo.	Il giorno di disponibilità è stato inserito correttamente.
2	Data con formato non valido.	Data con formato non valido [ERROR].  Elementi di sistema database validi.	Il database è inizializzato correttamente. Il maestro ha effettuato l'accesso.	{  Data: "09/07/24" }	Data inserita non esistente oppure formato errato.  Riprovare...	Il sistema chiede di inserire nuovamente la data.

3	Data già presente nel database.	Data con formato valido.  Data già presente nel database [ERROR].	Il database è inizializzato correttamente. Il maestro ha effettuato l'accesso.	{  Data: "09/10/2024" }	Giorno già presente.	Il caso d'uso termina notificando la presenza del giorno di disponibilità.
---	---------------------------------	---	--	----------------------------------	----------------------	--

## PIANO DI TEST UTILIZZANDO IL METODO DEL *CATEGORY-PARTITION TESTING* PER LA FUNZIONALITÀ “Modifica Giorno”

Data da modificare	Nuova data di disponibilità	Elemento di sistema database
<ul style="list-style-type: none"><li>• Data da modificare con formato valido (gg/mm/aaaa)</li><li>• Data da modificare con formato non valido [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• Nuova data di disponibilità con formato valido (gg/mm/aaaa)</li><li>• Nuova data di disponibilità con formato non valido [ERROR]</li></ul>	<ul style="list-style-type: none"><li>• Data da modificare presente nel database e nuova data disponibile.</li><li>• Data da modificare non presente nel database [ERROR]</li><li>• Nuova data non disponibile [ERROR]</li></ul>

Il numero di test da effettuarsi senza particolari vincoli è:  $2*2*3 = 12$

Introduciamo i vincoli [ERROR].

Il numero di test da eseguire per testare singolarmente i vincoli è 4 (1 per Data da modificare, 1 per Nuova data di disponibilità, 2 per Elemento di sistema database).

Il numero di test risultante è  $(1*1*1) + 4 = 5$

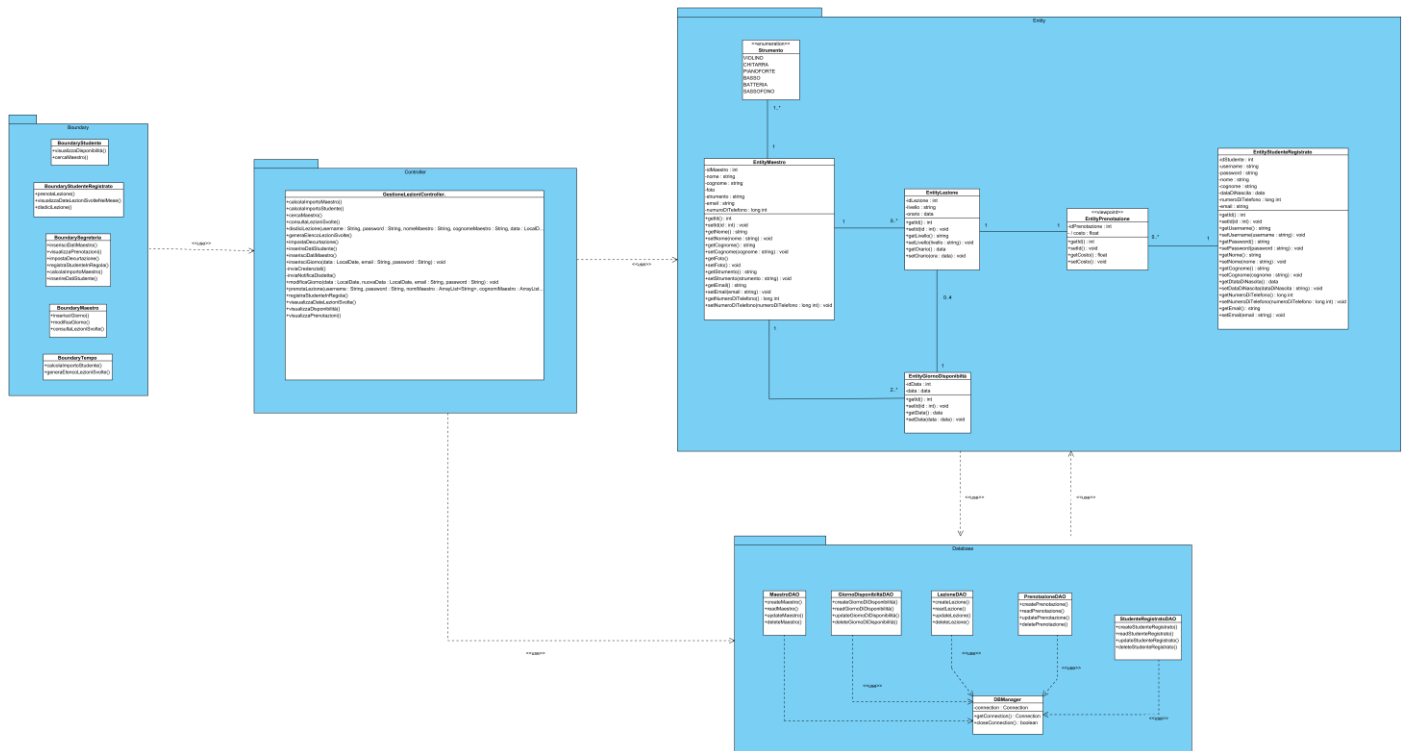
## TEST SUITE

Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Tutti gli input sono validi.	Data da modificare con formato valido.  Nuova data di disponibilità con formato valido.  Elementi di sistema database validi.	Il database è inizializzato correttamente. Il maestro ha effettuato l'accesso.	{  Data da modificare: "22/05/2025"  Nuova data: "25/05/2025"  }	Giorno di disponibilità modificato con successo!	La data del giorno di disponibilità selezionato è stata modificata con successo.
2	Data da modificare con formato non valido.	Data da modificare con formato non valido [ERROR].  Nuova data di disponibilità con formato valido.  Elementi di sistema database validi.	Il database è inizializzato correttamente.  Il maestro ha effettuato l'accesso.	{  Data da modificare: "22-05-2025"  Nuova data: "25/05/2025"  }	Data inserita non esistente oppure formato errato.  Riprovare...	Il sistema chiede di inserire nuovamente la data da modificare.
3	Nuova data di disponibilità con formato non valido.	Data da modificare con formato valido.  Nuova data di disponibilità	Il database è inizializzato correttamente.  Il maestro ha effettuato l'accesso.	{  Data da modificare: "22/05/2025"	Data inserita non esistente oppure formato errato.  Riprovare...	Il sistema chiede di reinserire la nuova data di disponibilità.

		con formato non valido [ERROR].  Elementi di sistema database validi.		Nuova data: "25-05-2025"  }		
4	Data da modificare non presente nel database.	Data da modificare con formato valido.  Nuova data di disponibilità con formato valido.  Data da modificare non presente nel database [ERROR].	Il database è inizializzato correttamente.  Il maestro ha effettuato l'accesso.	{  Data da modificare: "24/05/2025",  Nuova data: "25/05/2025"  }	Giorno indicato non presente nel sistema.	Il sistema chiede di inserire nuovamente la data da modificare.
5	Nuova data non disponibile.	Data da modificare con formato valido.  Nuova data di disponibilità con formato valido.  Nuova data di disponibilità già presente nel database [ERROR].	Il database è inizializzato correttamente.  Il maestro ha effettuato l'accesso.	{  Data da modificare: "22/05/2025"  Nuova data: "22/05/2025"  }	Nuovo giorno indicato già presente nel sistema.	Il sistema chiede di reinserire la nuova data di disponibilità.

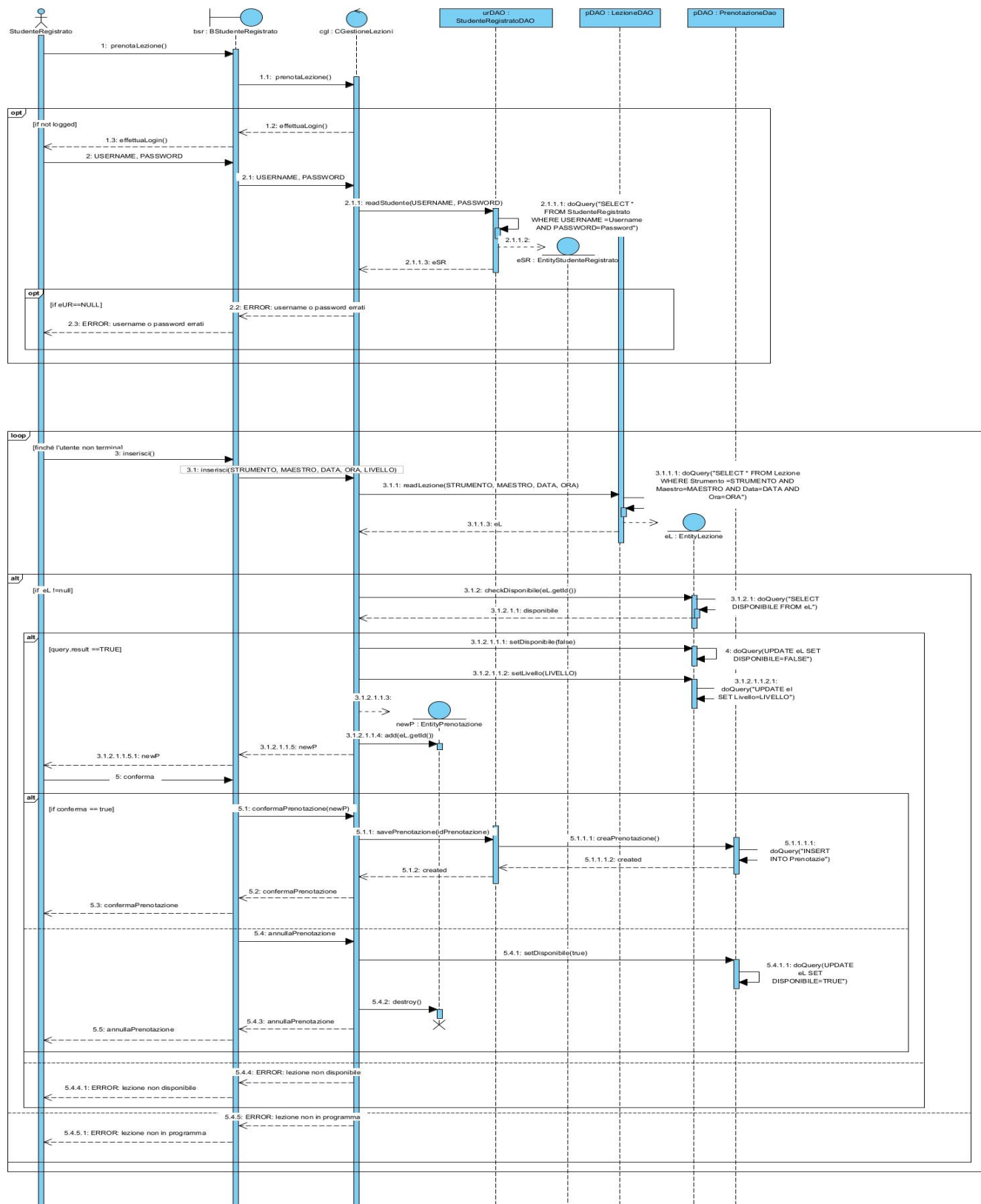
# 5 – Progettazione

## 5.1 – Diagramma delle classi

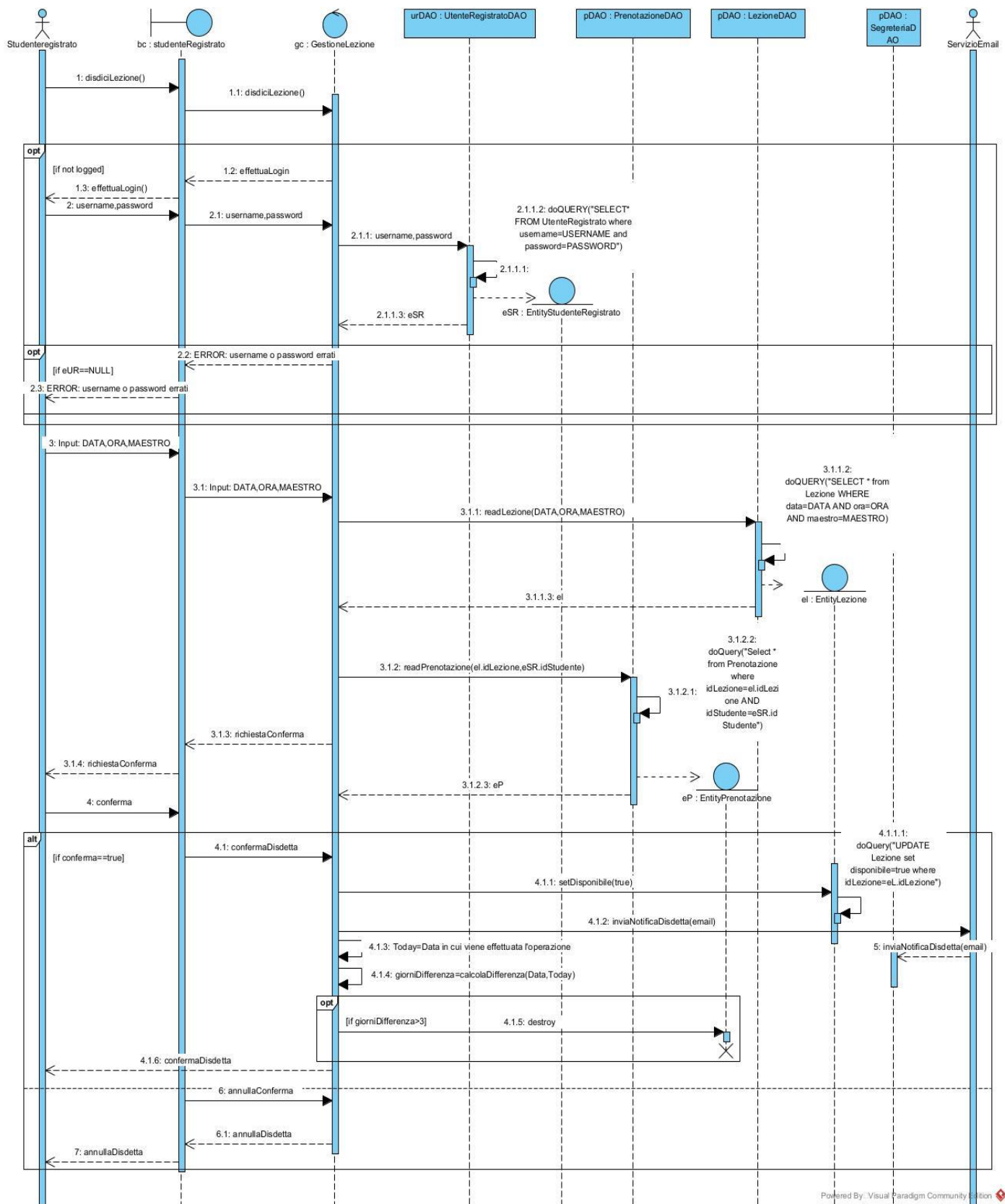


## 5.2 – Diagrammi di sequenza

### UC3: PrenotaLezione

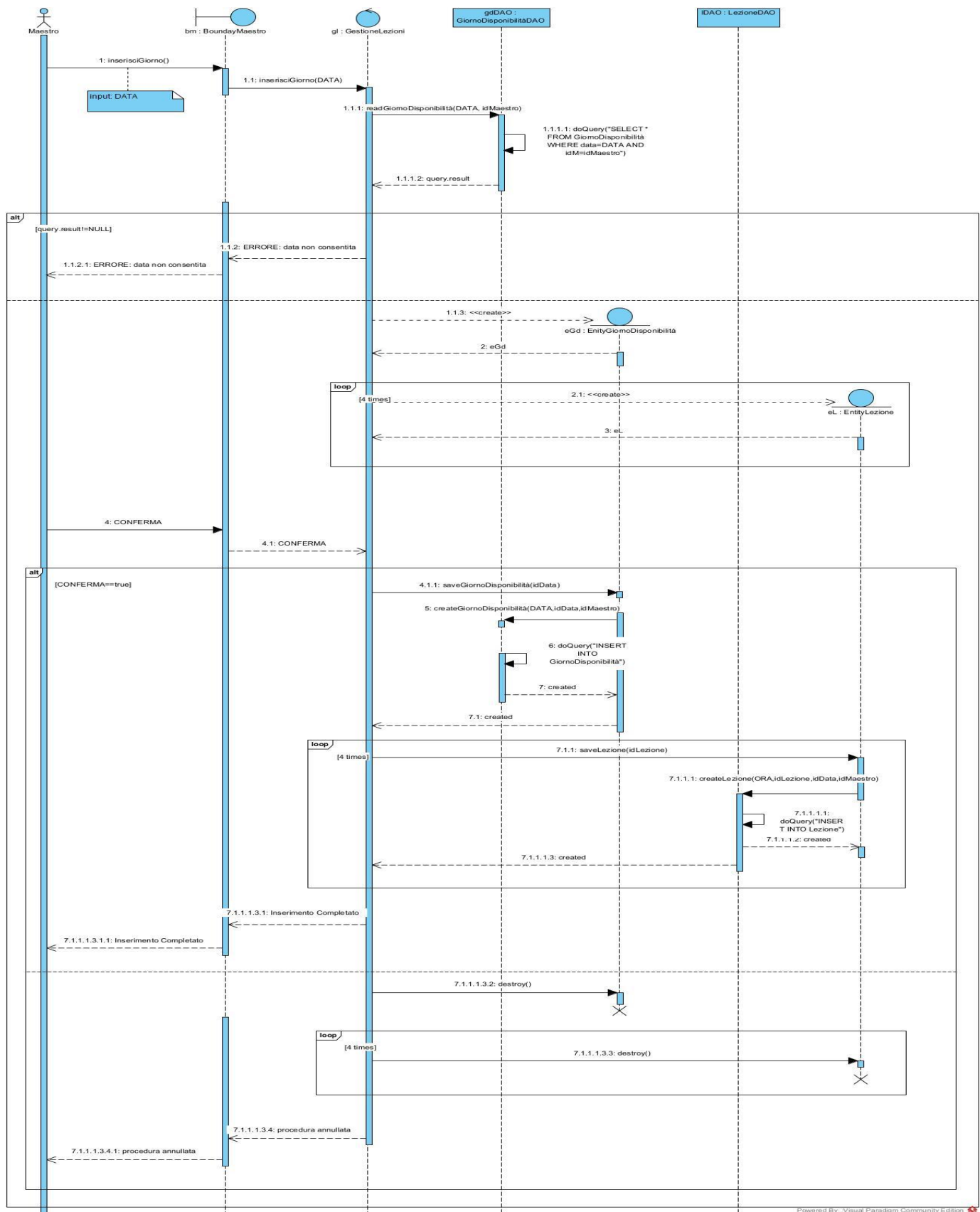


## UC4: DisdiciLezione

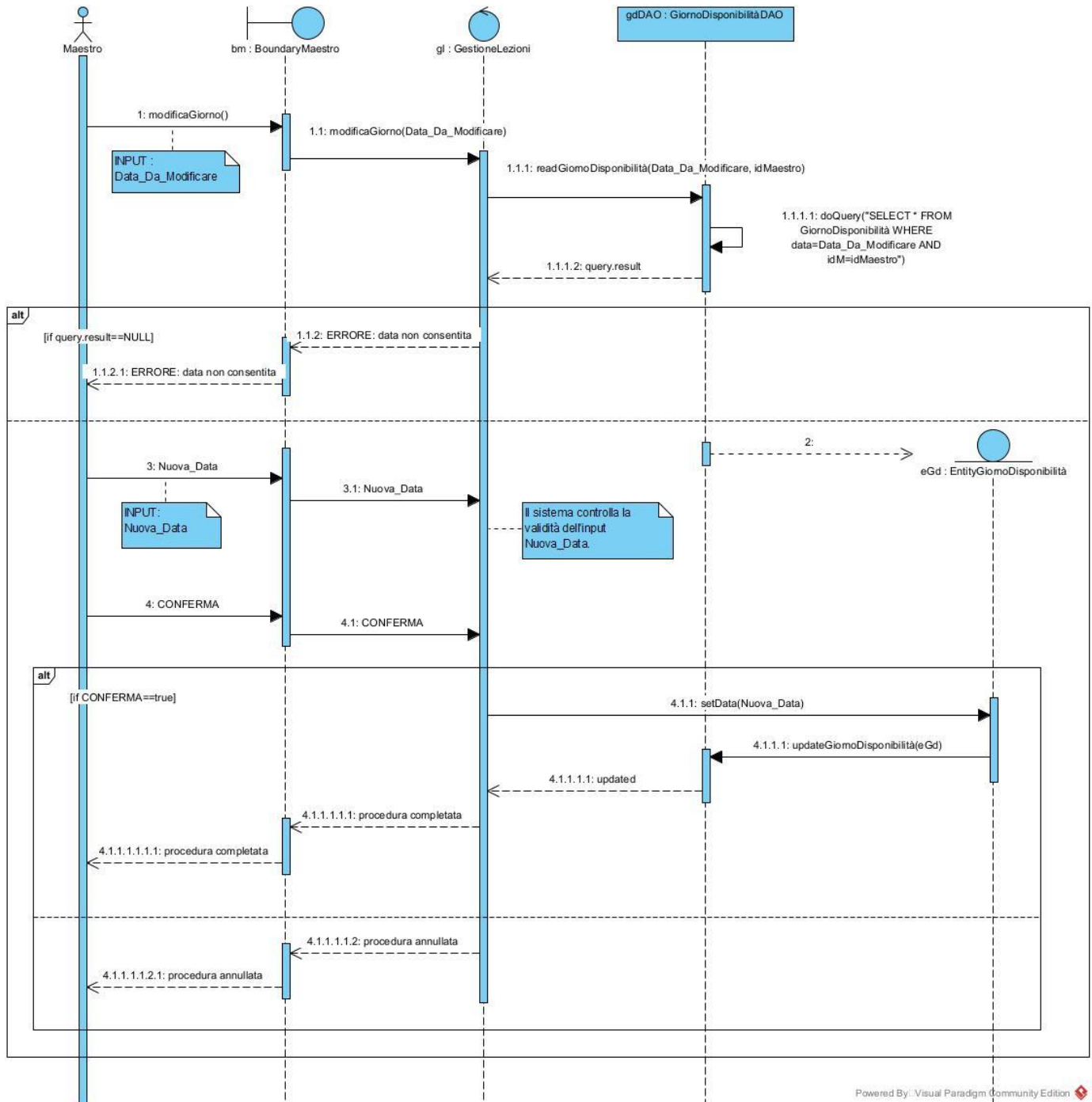




# UC6: InserisciGiorno



## UC7: ModificaGiorno



Powered By: Visual Paradigm Community Edition

## 6 – Implementazione

### Package:

- Control
- Boundary
- Entity
- DAO
- Exception
- Test

### Package Control:

#### Classi

- GestioneLezioni

#### Librerie Utilizzate

- import java.sql.Time;
- import java.time.LocalDate;
- import java.time.LocalTime;
- import java.time.temporal.ChronoUnit;
- import java.util.ArrayList;

### Package Boundary:

#### Classi

- BoundayStudenteRegistrato
- BoundayMaestro

#### Librerie Utilizzate

- import java.time.LocalDate;
- import java.time.LocalTime;
- import java.time.format.DateTimeFormatter;
- import java.util.ArrayList;
- import java.util.Scanner;
- import java.util.regex.Matcher;

- `import java.util.regex.Pattern;`

## **Package Entity:**

### Classi

- `Studente (abstract)`
- `StudenteRegistrato`
- `Maestro`
- `Lezione`
- `Prenotazione`
- `GiornoDisponibilità`

### Librerie Utilizzate

- `import java.time.LocalDateTime;`
- `import java.time.LocalDate;`
- `import java.sql.Date;`

## **Package DAO:**

### Classi

- `DBManager`
- `GiornoDisponibilitàDAO`
- `MaestroDAO`
- `StudenteRegistratoDAO`
- `LezioneDAO`
- `PrenotazioneDAO`

### Librerie Utilizzate

- `import java.sql.Connection;`
- `import java.sql.Date;`
- `import java.sql.PreparedStatement;`
- `import java.sql.ResultSet;`
- `import java.sql.SQLException;`
- `import java.time.LocalDate;`

- `import java.time.format.DateTimeFormatter;`
- `import java.sql.Time;`
- `import java.time.LocalTime;`
- `import java.util.ArrayList;`
- `import java.sql.DriverManager;`

## **Package Exception**

### **Classi**

- `DAOException`
- `DBConnectionException`
- `OperationException`

## **Artefatti necessari per l'installazione ed esecuzione del programma**

- `mysql-connector-j-8.3.0.jar`
- `junit-4.13.2-javadoc.jar`
- `junit-4.13.2.jar`
- `hamcrest-core-1.3-javadoc.jar`
- `hamcrest-core-1.3.jar`
- XAMPP
- MySQL Workbench 8.0.36

Nel caso venga utilizzato Visual Studio Code si consiglia di aggiungere le estensioni:

- Database Client JDBC
- JUnit JAR Downloader
- Extension Pack for Java

## 7 - Testing

### 7.1 – Test funzionale

#### FUNZIONALITÀ “Prenota Lezione”

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 gli input sono validi.	Username valido, Password valida, Maestro valido, Data valida, Ora valida, Strumento valido, Livello valido, Elementi di sistema database validi.	Il database è inizializzato correttamente. Lo studente effettua il login correttamente. Le lezioni sono disponibili per lo strumento nella data e ora selezionate.	{ Username: “anto24”, Password: “ciaorag a20”, Nome Maestro : “Andrea”, Cognome Maestro : “Pirlo”, Data: “24/06/2024”, Ora: “18:00”, Strumento:	Prenotazione avvenuta con successo.	Lo studente ottiene la sua prenotazione.	Prenotazione avvenuta con successo.	Lo studente ottiene la sua prenotazione.	PASS

				"Sassofono",  Livello: "Base" }					
2	Username: stringa > 50 caratteri.	Username: stringa > 50 caratteri [ERROR],  Validi:  Password, Maestro, Data, Ora, Strumento, Livello, ElementoDi SistemaDatabase.	Lo studente vuole accedere per effettuare la prenotazione.	{ Username: "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa",  Password: "noccop813",  Nome Maestro: "Andrea",  Cognome Maestro: "Pirlo",  Data: "24/06/2024",	Username non valido.	Il sistema chiede nuovamente di inserire l'username.	Username non valido.	Il sistema chiede nuovamente di inserire l'username.	PASS

				<p>Ora: "18:00",</p> <p>Strumento: "Sassofono",</p> <p>Livello: "Base"</p> <p>}</p>					
3	<p>Password: stringa di caratteri di lunghezza &gt; 50</p>	<p>Password: stringa di caratteri di lunghezza &gt;50 [ERROR],</p> <p>Validi:</p> <p>Username, Maestro, Data, Ora, Strumento, Livello, ElementiDiSistemaDatabase.</p>	<p>Lo studente registrato seleziona la funzionalità "Prenotazione Lezione".</p> <p>Il sistema richiede all'utente registrato l'inserimento della password.</p>	<p>{</p> <p>Username: "dottoraro2",</p> <p>Password: "htppp11",</p> <p>Nome Maestro: "Giancarlo",</p> <p>Cognome Maestro: "Zita",</p> <p>Data: "25/06/2024",</p> <p>Ora: "17:00",</p> <p>Strumento:</p>	<p>Password non valida.</p>	<p>Il sistema chiede nuovamente di inserire la password dell'utente.</p>	<p>Password non valida.</p>	<p>Il sistema chiede nuovamente di inserire la password.</p>	PASS



				“Batteria”,  Livello: “Base” }					
4	Password: stringa di caratteri di lunghezza ≤8	Password: stringa di caratteri di lunghezza >8 [ERROR],  Validi:  Username, Maestro, Data, Ora, Strumento, Livello, ElementiDiS istemaData base.	Lo studente registrato seleziona la funzionali tà prenota zione.  Il sistema richiede all’utente registrato l’inserime nto della password.	{ Userna me: “angy75 37”,  Passwor d: “FNS!19 26”,  Nome maestro : “Mauro ”  Cognom e maestro : “Sassi”,  Data: ”29/06/ 2024”,  Ora: “20:00”,  Strume nto: “sassofo no”,  Livello: “Base”	Passwo rd non valida .	Il sistema chiede nuovamen te di inserire la password dell’utente .	Passwo rd non valida.	Il sistema chiede nuovam ente di inserire la passwor d dell’ute nte.	PASS

				}					
5	Nome Maestro: stringa > 50 caratteri.	Maestro stringa > 50 caratteri [ERROR],  Validi:  Username, Password, Cognome Maestro, Data, Ora, Strumento, Livello, ElementoDi SistemaData base.	Lo studente vuole accedere per effettuare la prenotazi one.	{ Userna me: "aaaaaa aaaaaa" , Passwor d: "nocap po813",  Nome maestro : "Andrea aaaaaaa aaaaaaa aaaaaaa aaaaaaa aaaaaaa aaaaaaa aaaaaaa aaaaaaa aaaaaaa",  Cognom e Maestro : "Galimb erti",  Data: "24/06/ 2024",  Ora: "18:00",  Strume nto:	Nome Maestr o non valido.	Il sistema chiede nuovamen te di inserire il nome del maestro.	Nome maestr o non valido.	Il sistema chiede di inserire il nome del maestro .	PASS

				"Sassofono",  Livello: "Base" }					
6	Cognome Maestro: stringa > 50 caratteri.	Cognome Maestro stringa > 50 caratteri [ERROR],  Validi:  Username, Password, Nome Maestro, Data, Ora, Strumento, Livello, ElementoDi SistemaDatabase.	Lo studente vuole accedere per effettuare la prenotazione.	{ Username: "aaaaaa aaaaaa" , Password: "noca po813",  Nome Maestro: "Andrea",  Cognome Maestro: "Galimberti iiiiiiii iiiiiiii iiiiiiii iiiiiiii iiiiiiii iiiiiiii iiiiiiii iiiiiiii",  Data: "24/06/ 2024",	Cognome Maestro o non valido.	Il sistema chiede nuovamente di inserire il cognome del maestro.	Cognome maestro o non valido.	Il sistema chiede di inserire nuovamente il cognome del maestro.	PASS

				Ora: "18:00",  Strumento: "Sassofono",  Livello: "Base" }					
7	Data con formato non valido.	Data formato non valido [ERROR],  Validi:  Username, Password, Maestro, Ora, Strumento, Livello, ElementoDi SistemaDatabase.	Lo studente vuole effettuare la prenotazione.  Il sistema richiede all'utente l'inserimento della data.	{ Username: "oaksak oaos",  Password: "nocap po813",  Nome Maestro : "Mario",  Cognome Maestro : "Rossi",  Data: "24/06/24",  Ora: "14:00",  Strumento:	Data inserita non esiste oppure formato errato.  Riprova re...	Il sistema chiede di inserire nuovamente la data.	Data inserita non esiste oppure formato errato.  Riprova re...	Il sistema chiede di inserire nuovamente la data	PASS

				"Sassofono",  Livello: "Base" }					
8	Ora con formato non valido.	Ora con formato non valido [ERROR],  Validi:  Username, Password, Maestro, Data, Strumento, Livello, ElementoDi SistemaDataBase.	Lo studente vuole effettuare la prenotazione.  Il sistema richiede all'utente l'inserimento dell'ora.	{ Username: "oaksakaoas", Password: "nocapop813", Nome Maestro: "Mario", , Cognome Maestro: "Rossi",  Data: "24/06/2024",  Ora: "19",  Strumento: "Sassofono",  Livello: "Base"	Non ci sono lezioni per l'orario scelto. Le lezioni iniziano solo alle 17:00, alle 18:00, alle 19:00 oppure alle 20:00. Riprovare...	Il sistema chiede nuovamente di inserire l'ora.	Non ci sono lezioni per l'orario scelto. Le lezioni iniziano solo alle 17:00, alle 18:00, alle 19:00 oppure alle 20:00. Riprovare...	Il sistema chiede nuovamente di inserire l'ora.	PASS

				}					
9	Ora non compresa tra le 17.00 e le 21.00.	Ora non compresa tra le 17.00 e le 21.00 [ERROR],  Ora con formato valido,  Validi:  Username, Password, Maestro, Data, Strumento, Livello, ElementoDi SistemaDatabase.	Lo studente vuole effettuare la prenotazione.  Il sistema richiede all'utente l'inserimento dell'ora.	{ Userame: "oaksak oaos",  Passwor d: "nocap po813",  Nome Maestro : "Mario",  Cognom e Maestro : "Rossi",  Data: "24/06/ 2024",  Ora: "14:00",  Strume nto: "Sassof ono",  Livello: "Base" }	Non ci sono Lezioni per l'orario scelto. Le Lezioni iniziano solo alle 17:00, alle 18:00, alle 19:00 oppure alle 20:00. Riprova re...	Il sistema chiede nuovamen te di inserire l'ora.	Non ci sono Lezioni per l'orario scelto. Le Lezioni iniziano solo alle 17:00, alle 18:00, alle 19:00 oppure alle 20:00. Riprova re...	Il sistema chiede nuovam ente di inserire l'ora	PASS
10	Strumento non valido	Strumento non valido [ERROR],	Lo studente vuole	{ Userna me:	Non vengon o	Il sistema chiede di inserire	Non vengon o	Il sistema chiede	PASS

		Validi:  Username, Password, Maestro, Data, Ora, Livello, ElementoDi SistemaData base.	effettuare la prenotazi one.  Il sistema richiede all'utente l'inserime nto dello strumento .	"oaksak oaos",  Passwor d: "nocap po813",  Nome Maestro : "Mario" ,  Cognom e Maestro : "Rossi",  Data: "24/06/ 2024",  Ora: "18:00",  Strume nto: "Clarine tto",  Livello: "Base" }	erogate lezioni per lo Strume nto scelto.  Riprova re sceglie ndo un altro strume nto...	nuovamen te lo strumento	erogate lezioni per lo Strume nto scelto.  Riprova re sceglie ndo un altro strume nto...	di inserire nuovam ente lo strumen to	
11	Altri livelli.	Livello non valido [ERROR],  Validi:	Il livello della lezione inserito non è presente nel sistema.	{ Userna me: "salvo1 Ora",  Passwor d:	Le lezioni posson o essere solo di 2 livelli: BASE	Il sistema chiede nuovamen te di inserire il livello della lezione.	Le lezioni posson o essere solo di 2 livelli: BASE	Il sistema chiede nuovam ente di inserire il livello	PASS

		Username, Password, Maestro, Data, Ora, Strumento, ElementiDiS istemaData base.		“noleg3 9day”,  Nome Maestro : “Ludovi co”,  Cognom e Maestro : “Lemm e”,  Data: ”14/06/ 2024”,  Ora: “18:00”,  Strume nto: “Chitarr a”,  Livello: “Avanza to” }	(costo di 15.00 euro) oppure INTER MEDIO (costo di 30.00 euro)		(costo di 15.00 euro) oppure INTER MEDIO (costo di 30.00 euro)	della lezione	
12	Lezione non presente nel database [ERROR]	ElementoDi SistemaData base non valido [ERROR],  Validi:  Username, Password,	La lezione non risulta nel database.	{ Userna me: “lory44 6”,  Passwor d: “circuit o don”,	Lezione non trovata	Il sistema non trova lezione e non può procedere per la prenotazio ne.	Lezione non trovata	Il sistema non trova la lezione non può procede re per la prenota zione	PASS



		Maestro, Data, Ora, Strumento, Livello.		Nome Maestro : "Andrea",  Cognom e Maestro : "Presti",  Data: "13/06/ 2024",  Ora: "21:00",  Strume nto: "Batteri a",  Livello: "Interm edio" }					
13	Lezione già prenotata [ERROR]	ElementoDi SistemaData base non valido [ERROR],  Validi:  Username, Password, Maestro, Data, Ora, Strumento,	La lezione risulta essere già prenotata.	{ Userna me: "pablo7 2",  Passwor d: "aitriple i22",  Nome Maestro :	Lezione non disponi bile.	Il sistema non può prenotare la lezione. .	La lezione non è disponi bile.	Il sistema non può prenota re la lezione.	PASS

		Livello.		“Stefano”,  Cognome Maestro : “Pioli”,  Data: ”26/06/ 2024”,  Ora: “19:00”,  Strumento: “Pianoforte”,  Livello: “Base” }					
14	Username e/o password non presenti nel Database [ERROR]	ElementoDi SistemaData base non valido [ERROR],  Validi:  Username, Password, Maestro, Data, Ora, Strumento, Livello.	Username e/o password utente non sono presenti nel Database	{ Username: “manu23”,  Password: “Maestro registrato”,  Nome maestro : “Mario” ,	Studente non registrato.	Il sistema non può prenotare la lezione.	Studente non registrato.	Il sistema non può prenotare la lezione.	PASS

				Cognom e maestro : "Rossi",  Data: "22/07/ 2024",  Ora: "17:00",  Strume nto: "Sassof ono",  Livello: "Interm edio" }						
--	--	--	--	--	--	--	--	--	--	--

## FUNZIONALITÀ “Disdici Lezione”

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 gli input sono validi.	Maestro valido,  Data valida,  Ora valida,  ElementoDiSistemaDatabase valido.	Lo studente ha effettuato il login.  Il database è inizializzato correttamente. La lezione esiste per la data, ora e per il maestro selezionato.	{ Nome Maestro : “Andrea”,  Cognome Maestro : “Pirlo”,  Data: “27/06/2024”,  Ora: “19.00” }	DISDETTA AVVENUTA CON SUCCESSO!	La lezione è stata disdetta correttamente.	DISDETTA AVVENUTA CON SUCCESSO!	La lezione è stata disdetta correttamente.  Il sistema notifica via mail la disdetta alla segreteria	PASS
2	Nome Maestro: stringa > 50 caratteri	Nome Maestro stringa > 50 caratteri [ERROR],  Validi:  Data, Ora, ElementoDiSistemaDatabase.	Lo studente ha inserito il maestro per la disdetta della lezione.	{ Nome Maestro : “Andreaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa”,  Cognome Maestro : “Pirlo” }	Nome Maestro non valido.	Il sistema chiede nuovamente di inserire il Nome maestro.	Nome maestro non valido.	Il sistema chiede nuovamente di inserire il Nome maestro	PASS

				Data: "24/06/ 2024",  Ora: "18:00" }					
3	Cognome Maestro: stringa > 50 caratteri	Cognome Maestro stringa > 50 caratteri [ERROR],  Validi:  Data, Ora, Nome Maestro, ElementoDiS istemaDatab ase.	Lo studente ha inserito il maestro per la disdetta della lezione.	{ Nome Maestro : "Andrea ",  Cognom e Maestro : "Pirlooo oooooo oooooo oooooo oooooo oooooo oooooo oooooo oooooo oooooo "  Data: "24/06/ 2024",  Ora: "18:00" }	Cogno me maestr o non valido	Il sistema chiede di inserire nuovam ente il cognom e del maestro	Cognom e maestro non valido	Il sistema chiede di inserire nuovame nte il cognome del maestro	PASS
4	Data con formato	Data con formato non	Lo studente	{ Nome	Data inserita	Il sistema	Data	Il sistema	PASS

	non valido.	valido [ERROR],  Validi:  Maestro, Ora, ElementoDiSistemaDatabase.	vuole effettuare la disdetta.	Maestro : "Mario",  Cognome Maestro : "Rossi",  Data: "24/06/24",  Ora: "18:00"  }	non valida. Riprovare...	chiede nuovamente di inserire la data.	Riprova e...	chiede nuovamente di inserire la data	
5	Ora con formato non valido.	Ora con formato non valido [ERROR],  Validi:  Maestro, Data, ElementoDiSistemaDatabase.	Lo studente vuole effettuare la disdetta.	{ Nome Maestro : "Mario",  Cognome Maestro : "Rossi",  Data: "24/06/2024",  Ora: "18"  }	Non ci sono Lezioni per l'orario scelto. Le Lezioni iniziano solo alle 17:00, alle 18:00, alle 19:00 oppure alle 20:00. Riprovare...	Il sistema chiede nuovamente di inserire l'ora.	Non ci sono Lezioni per l'orario scelto.  Le Lezioni iniziano solo alle 17:00, alle 18:00, alle 19:00 oppure alle 20:00.  Riprova e...	Il sistema chiede nuovamente di inserire l'ora.	PASS

6	Ora non compresa tra le 17.00 e le 21.00.	Ora non compresa tra le 17.00 e le 21.00 [ERROR],  Validi:  Maestro, Data, ElementoDiSistemaDatabase.	Lo studente vuole effettuare la disdetta.	{ Nome Maestro : "Mario" Cognome Maestro : "Rossi", Data: "24/06/2024", Ora: "14.00" }	Non ci sono Lezioni per l'orario scelto. Le Lezioni iniziano solo alle 17:00, alle 18:00, alle 19:00 oppure alle 20:00. Riprovare...	Il sistema chiede nuovamente di inserire l'ora.	Non ci sono Lezioni per l'orario scelto. Le Lezioni iniziano solo alle 17:00, alle 18:00, alle 19:00 oppure alle 20:00.  Riprovarle...	Il sistema chiede nuovamente di inserire l'ora.	PASS
7	Lezione non presente nel database [ERROR].	ElementoDiSistemaDatabase non valido [ERROR],  Validi:  Maestro, Data, Ora.	La lezione non è disponibile .	{ Nome Maestro : "Stefano", Cognome Maestro : "Pioli", Data: "26/06/2024", Ora: "19:00" }	Lezione non trovata	Il sistema non può effettuare la disdetta per la lezione richiesta .	Lezione non trovata.	Il sistema non può effettuare la disdetta per la lezione richiesta.	PASS

8	Lezione non prenotata [ERROR]	ElementoDiSistemaDatabase non valido [ERROR],  Validi:  Maestro, Ora, Data.	La lezione non è stata prenotata.	{ Nome Maestro : "Stefano",  Cognome Maestro : "Pioli",  Data: "26/06/2024",  Ora: "19:00" }	Lezione non trovata .	Il sistema non può effettuare la disdetta per la lezione richiesta .	Lezione non trovata.	Il sistema non può effettuare la disdetta per la lezione richiesta.	PASS
---	-------------------------------	---	-----------------------------------	--	-----------------------	--	----------------------	---	------



## FUNZIONALITÀ “Inserisci Giorno”

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 gli input sono validi.	Data con formato valido.  Elementi di sistema database validi.	Il database è inizializzato correttamente.  Il maestro ha effettuato l'accesso.	{  Data: "09/07/2024"  }	Un nuovo giorno di disponibilità è stato inserito con successo.	Il giorno di disponibilità è stato inserito correttamente.	Un nuovo giorno di disponibilità è stato inserito con successo.	Il giorno di disponibilità è stato inserito correttamente.	PASS
2	Data con formato non valido.	Data con formato non valido [ERROR].  Elementi di sistema database validi.	Il database è inizializzato correttamente.  Il maestro ha effettuato l'accesso.	{  Data: "09/07/24"  }	Data inserita non esistente oppure formato errato.  Riprova re...	Il sistema chiede di inserire nuovamente la data.	Data inserita non esistente oppure formato errato.  Riprova re...	Il sistema chiede di inserire nuovamente la data.	PASS
3	Data già presente nel database.	Data con formato valido.  Data già presente nel database [ERROR].	Il database è inizializzato correttamente.  Il maestro ha effettuato l'accesso.	{  Data: "09/10/2024"  }	Giorno già presente.	Il caso d'uso termina notificando la presenza del giorno di disponibilità.	Giorno già presente.	Il caso d'uso termina notificando la presenza del giorno di disponibilità.	PASS

--	--	--	--	--	--	--	--	--	--

## FUNZIONALITÀ “Modifica Giorno”

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 gli input sono validi.	Data da modificare con formato valido.  Nuova data di disponibilità con formato valido.  Elementi di sistema database validi.	Il database è inizializzato o correttamente.  Il maestro ha effettuato l'accesso.	{  Data da modificare: "22/05/2025" ,  Nuova data: "25/05/2025" }	Giorno di disponibilità modificato con successo!  	La data del giorno di disponibilità selezionata o è stata modificata con successo.	Giorno di disponibilità modificato con successo!	La data del giorno di disponibilità selezionata o è stata modificata con successo.	PASS
2	Data da modificare con formato non valido.	Data da modificare con formato non valido [ERROR].  Nuova data di disponibilità con formato valido.  Elementi di sistema database validi.	Il database è inizializzato o correttamente.  Il maestro ha effettuato l'accesso.	{  Data da modificare: "22-05-2025",  Nuova data: "25/05/2025" }	Data inserita non esistente oppure formato errato.  Riprova re...	Il sistema chiede di inserire nuovamente la data da modificare.  	Data inserita non esistente oppure formato errato.  Riprova re...	Il sistema chiede di inserire nuovamente la data da modificare.	PASS
3	Nuova data di disponibilità con formato	Data da modificare con formato valido.  Nuova data di disponibilità con	Il database è inizializzato	{  Data da modificare:	Data inserita non esistente oppure	Il sistema chiede di reinserire la nuova data di	Data inserita non esistente oppure	Il sistema chiede di reinserire la nuova data di	PASS

	o non valido.	formato non valido [ERROR].  Elementi di sistema database validi.	correttamente.  Il maestro ha effettuato l'accesso.	"22/05/2025" ,  Nuova data: "25-05-2025"  }	formato errato.  Riprova re...	disponibilità.	formato errato.  Riprova re...	disponibilità.	
4	Data da modificare non presente nel database.	Data da modificare con formato valido.  Nuova data di disponibilità con formato valido.  Data da modificare non presente nel database [ERROR].	Il database è inizializzato correttamente.  Il maestro ha effettuato l'accesso.	{  Data da modificare: "24/05/2025" ,  Nuova data: "25/05/2025" }	Giorno indicato non presente nel sistema.	Il sistema chiede di inserire nuovamente la data da modificare.	Giorno indicato non presente nel sistema.	Il sistema chiede di inserire nuovamente la data da modificare.	PASS
5	Nuova data di disponibilità già presente nel database.	Data da modificare con formato valido.  Nuova data di disponibilità con formato valido.  Nuova data di disponibilità già presente nel database [ERROR].	Il database è inizializzato correttamente.  Il maestro ha effettuato l'accesso.	{  Data da modificare: "22/05/2025" ,  Nuova data: "22/05/2025" }	Nuovo giorno indicato o già presente nel sistema.	Il sistema chiede di reinserire la nuova data di disponibilità.	Nuovo giorno indicato già presente nel sistema.	Il sistema chiede di reinserire la nuova data di disponibilità.	PASS

