

Università degli Studi di Salerno
Corso di Ingegneria del Software

EduCat
Test Case Specification Document
Versione 1.0



EduCat

Data: 21/12/2025

Progetto: EduCat	Versione: 1.0
Documento: Test Case Specification	Data: 21/12/2025

Coordinatore del progetto:

Nome	Matricola
Isabella Dima	0512119719

Partecipanti:

Nome	Matricola
Isabella Dima	0512119719
Anna Chiara Esposito	0512119143

Scritto da:	Anna Chiara Esposito
-------------	----------------------

Revision History

Data	Versione	Descrizione	Autore
21/12/2025	1.0	Test Case Specification	Tutto il team

Indice

1.	Introduction	5
2.	TC_LEZ_01: Prenotazione con successo	5
2.1.	Test case specification identifier	5
2.2.	Test items	5
2.3.	Input Specifications.....	5
2.4	Output Specifications	5
2.5	Environmental Needs	5
2.6	Special procedural requirements	5
2.7	Intercase dependencies	5
3.	TC_LEZ_02: Tentativo di prenotazione su slot già occupato	5
3.1.	Test case specification identifier	5
3.2.	Test items	6
3.3.	Input Specifications.....	6
3.4.	Output Specifications	6
3.5.	Environmental Needs	6
3.6.	Special procedural requirements	6
3.7.	Intercase dependencies.....	6
4.	TC_LEZ_03: Prenotazione con dati di pagamento errati	6
4.1.	Test case specification identifier	6
4.2.	Test items	6
4.3.	Input Specifications.....	6
4.4.	Output Specifications	7
4.5.	Environmental Needs	7
4.6.	Special procedural requirements	7
4.7.	Intercase dependencies.....	7
5.	TC_LEZ_04: Verifica eccezione “SlotOccupatoException”.....	7
5.1.	Test case specification identifier	7
5.2.	Test items	7
5.3.	Input Specifications.....	7
5.4.	Output Specifications	7
5.5.	Environmental Needs	7
5.6.	Special procedural requirements	7
5.7.	Intercase dependencies.....	8
6.	TC_LEZ_05: Rollback della transazione Database in caso di fallimento pagamento	8
6.1.	Test case specification identifier	8
6.2.	Test items	8
6.3.	Input Specifications.....	8
6.4.	Output Specifications	8
6.5.	Environmental Needs	8
6.6.	Special procedural requirements	8
6.7.	Intercase dependencies.....	8
7.	TC_LEZ_06: Gestione Concorrenza (Race Condition)	8
7.1.	Test case specification identifier	8
7.2.	Test items	8
7.3.	Input Specifications.....	8

7.4.	Output Specifications	8
7.5.	Environmental Needs	8
7.6.	Special procedural requirements	8
7.7.	Intercase dependencies	8
8.	TC_LEZ_07: Copertura del path “Annullamento Lezione” con rimborsi	8
8.1.	Test case specification identifier	8
8.2.	Test items	8
8.3.	Input Specifications.....	8
8.4.	Output Specifications	8
8.5.	Environmental Needs	9
8.6.	Special procedural requirements	9
8.7.	Intercase dependencies	9

1. Introduction

Questo documento specifica i casi di test per il sottosistema "Gestione Lezione", focalizzandosi sulla funzionalità critica di Prenotazione e Acquisto Lezione. I test coprono sia il flusso corretto che i flussi alternativi di errore, verificando l'integrazione tra l'interfaccia utente, la logica di business e la persistenza dei dati.

2. TC_LEZ_01: Prenotazione con successo

2.1. Test case specification identifier

- TC_LEZ_01: Prenotazione con successo

2.2. Test items

- it.unisa.educat.view.GestioneLezioneGUI (Pagina di Checkout)
- it.unisa.educat.control.PrenotazioneServlet (Controller)
- it.unisa.educat.business.GestioneLezioneManager (Metodo prenotaLezione)
- it.unisa.educat.storage.GestioneLezioneDAO

2.3. Input Specifications

- Precondizione: l'utente è loggato

seleziona la lezione e inserisce i dati di pagamento (crearli per l'esempio)

2.4 Output Specifications

- Output Utente: Reindirizzamento alla pagina "Le mie lezioni" con messaggio di successo "Prenotazione confermata".
- Stato Sistema: Nel Database, tabella Prenotazione, inserita nuova riga con IDStudente=101, IDLezione=50, stato="attiva".

2.5 Environmental Needs

- Server: Apache Tomcat 10.1 in esecuzione.
- Database: MySQL 8.0 con dataset di test caricato (l'ID Lezione 50 deve esistere ed essere libero).
- Browser/Driver: Google Chrome / Selenium WebDriver per test di sistema; JUnit 5 per test di integrazione.

2.6 Special procedural requirements

Nessuno. Il test può essere eseguito in qualsiasi momento purché il database sia nello stato startup previsto.

2.7 Intercase dependencies

Dipende dal Test Case TC_AUTH_01 (Login Studente): l'utente deve avere una sessione attiva per poter procedere al checkout.

3. TC_LEZ_02: Tentativo di prenotazione su slot già occupato

3.1. Test case specification identifier

Educat	Ingegneria del Software	Pagina 5 di 9
--------	-------------------------	---------------

- TC_LEZ_02: Tentativo di prenotazione su slot già occupato

3.2. Test items

Verifica la gestione delle eccezioni di business nel Manager e la loro presentazione all'utente.

- it.unisa.educat.business.GestioneLezioneManager
- Eccezione attesa: SlotOccupatoException (come definito nell'Object Design).

3.3. Input Specifications

- Precondizione: l'utente è loggato
- Lezione selezionata (id) e dati di pagamento

3.4. Output Specifications

- Output Utente: La pagina di checkout viene ricaricata mostrando un messaggio di errore: "*Errore: Lo slot selezionato non è più disponibile.*"
- Stato Sistema: Nessuna nuova riga inserita nella tabella Prenotazione. Lo stato del database rimane invariato.

3.5. Environmental Needs

- Database: MySQL 8.0 con dataset di test caricato (l'ID Lezione 50 deve esistere ed essere libero).
- Browser/Driver: Google Chrome / Selenium WebDriver per test di sistema; JUnit 5 per test di integrazione.
- Vincolo Dati: Richiede che il database contenga già una prenotazione attiva per la Lezione ID xx (condizione creata dall'esecuzione di TC_LEZ_01).

3.6. Special procedural requirements

Nessuno. Il test può essere eseguito in qualsiasi momento purché il database sia nello stato startup previsto.

3.7. Intercase dependencies

Dipende da TC_LEZ_01 Questo test deve essere eseguito **dopo** che il test TC_LEZ_01 ha occupato lo slot con successo, oppure richiede una pre-condizione manuale sul DB.

4. TC_LEZ_03: Prenotazione con dati di pagamento errati

4.1. Test case specification identifier

- TC_LEZ_03: Prenotazione con dati di pagamento errati

4.2. Test items

Verifica i controlli di validazione input nel Controller/Servlet prima di invocare il Manager

- it.unisa.educat.control.PrenotazioneServlet

4.3. Input Specifications

- Precondizione: l'utente è loggato
- Lezione selezionata e dati di pagamento errati (numero di carta troppo corto)

4.4. Output Specifications

- Output Utente: Messaggio di errore inline nel form: "*Dati di pagamento non validi. Controllare numero carta e CVV.*"
- Stato Sistema: Il metodo GestioneLezioneManager.prenotaLezione() non viene invocato. Nessuna modifica al DB.

4.5. Environmental Needs

- Apache Tomcat 10.1
- JUnit per invocare la Servlet in ambiente mock.

4.6. Special procedural requirements

Nessuno. Il test può essere eseguito in qualsiasi momento purché il database sia nello stato startup previsto.

4.7. Intercase dependencies

Nessuna dipendenza specifica da altri test case funzionali.

5. TC_LEZ_04: Verifica eccezione “SlotOccupatoException”

5.1. Test case specification identifier

- TC_LEZ_04: Verifica eccezione “SlotOccupatoException”

5.2. Test items

Verifica la logica condizionale del rimborso nel Manager.

- it.unisa.educat.business.GestioneLezioneManager (Metodo annullaLezione)

5.3. Input Specifications

- Data Lezione: 20/01/2026
- Data Corrente (Simulata dal sistema): 18/01/2026 (Differenza > 24 ore).
- Azione: Click su "Annulla Lezione".

5.4. Output Specifications

- Output Utente: Messaggio "Lezione annullata con successo. Il rimborso è stato emesso."
- Stato Sistema:
 - Tabella Prenotazione: stato aggiornato a “annullata”.
 - Tabella Pagamento: stato aggiornato a “rimborsato”

5.5. Environmental Needs

Necessità di poter manipolare la data di sistema o di utilizzare librerie come Mockito per mockare LocalDate.now() durante il test di unità, al fine di soddisfare la condizione if (dataLezione - dataCorrente > 24h).

5.6. Special procedural requirements

Richiede l'intervento del tester (o dello script di test) per impostare la data corrente simulata prima dell'esecuzione.

5.7. Intercase dependencies

Richiede una prenotazione attiva esistente (creata da TC_LEZIONE_01 o inserita manualmente).

6. TC_LEZ_05: Rollback della transazione Database in caso di fallimento pagamento

6.1. Test case specification identifier

6.2. Test items

6.3. Input Specifications

6.4. Output Specifications

6.5. Environmental Needs

6.6. Special procedural requirements

6.7. Intercase dependencies

7. TC_LEZ_06: Gestione Concorrenza (Race Condition)

7.1. Test case specification identifier

7.2. Test items

7.3. Input Specifications

7.4. Output Specifications

7.5. Environmental Needs

7.6. Special procedural requirements

7.7. Intercase dependencies

8. TC_LEZ_07: Copertura del path “Annullamento Lezione” con rimborsi

8.1. Test case specification identifier

8.2. Test items

8.3. Input Specifications

8.4. Output Specifications

Educat	Ingegneria del Software	Pagina 8 di 9
--------	-------------------------	---------------

8.5. Environmental Needs

8.6. Special procedural requirements

8.7. Intercase dependencies