

Università degli Studi di Salerno
Corso di Ingegneria del Software

**EduCat
Test Plan Document
Versione 2.0**



EduCat

Data: 21/01/2026

Progetto: EduCat	Versione: 2.0
Documento: Test Plan	Data: 21/01/2026

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 Plan Document	Tutto il team
30/12/2025	1.1	Aggiunta Tecnica Category Partition	Tutto il team
21/01/2026	2.0	Aggiustati Test Cases	Tutto il team

Indice

1.	Introduction	4
2.	Relationship to other documents	4
3.	System overview.....	4
4.	Features to be tested/not to be tested	4
5.	Pass/Fail criteria	5
6.	Approach	5
6.1	Testing di Unità (White-Box).....	5
6.2	Testing di Integrazione	6
6.3	Testing di Sistema (Black-Box)	6
7.	Suspension and resumption	6
8.	Testing materials (hardware/software requirements).....	6
9.	Test cases	6
9.1	Testing di Sistema (Black-Box)	6
9.2.	Applicazione Tecnica Category Partition.....	7
9.2.1.	TC_LEZ_01 Prenotazione	8
9.2.2.	TC_LEZ_02 Nuova lezione tutor	10
9.2.3.	TC_LEZ_03 Annullamento lezione (Tutor).....	11
9.2.4.	TC_LEZ_04 Annullamento lezione (Studente)	12
9.2.5.	TC_LEZ_05 Login.....	13
9.2.6.	TC_LEZ_06 Registrazione	13
9.2.7.	TC_LEZ_07 Ricerca lezione	15
9.3	Testing Strutturale (White-Box).....	15
9.3.1	WB_TC_01 Nuova lezione tutor	16
9.3.2	WB_TC_02 Prenotazione lezione.....	17
9.3.3	WB_TC_03 Annullamento Prenotazione.....	17
9.3.4	WB_TC_04 Fascia oraria lezione (Tutor).....	17
9.3.5	WB_TC_05 Ricerca con Criteri.....	18
9.3.6	WB_TC_06 Mapping ResultSet	18
9.3.7	WB_TC_07 Inserimento Utente	18
9.3.8	WB_TC_08 Recupero per Email	19
9.3.9	WB_TC_09 Recupero per ID	19
9.3.10	WB_TC_10 Recupero tutti Utenti	20
9.3.11	WB_TC_11 Aggiornamento Utente.....	20
9.3.12	WB_TC_12 Eliminazione Utente	20
9.3.13	WB_TC_13 Mapping ResultSet a Utente.....	21

1. Introduction

Il presente documento descrive la strategia di testing pianificata per la piattaforma EduCat. L'obiettivo è definire un quadro di riferimento per la verifica e la validazione del software, assicurando che soddisfi i requisiti funzionali e non funzionali specificati.

Il testing mira a rilevare difetti nel codice e nella logica di business, garantendo che le funzionalità critiche come la prenotazione delle lezioni e la gestione dei pagamenti operino correttamente prima del rilascio finale previsto per gennaio 2026.

2. Relationship to other documents

Questo piano di test è strettamente correlato alla documentazione di progetto prodotta nelle fasi precedenti.

- **Requirements Analysis Document (RAD):** I test funzionali (System Testing) sono progettati per verificare gli scenari e i requisiti funzionali descritti nel RAD. Ogni *Test Case* funzionale è tracciabile verso uno specifico requisito (es. Acquisto Lezione).
- **System Design Document (SDD):** L'architettura del sistema (MVC, Java, MySQL) definita nel SDD guida la strategia di *Integration Testing* e la configurazione dell'ambiente di test.
- **Object Design Document (ODD):** La struttura delle classi e dei pacchetti definita nell'ODD costituisce la base per il *Unit Testing*.

3. System overview

Il sistema EduCat è basato su un'architettura Three-Tier (a tre livelli) che implementa il design pattern MVC (Model-View-Controller) su infrastruttura Client/Server. I componenti principali oggetto di test sono:

1. **Gestione Utente:** Moduli per registrazione, login e gestione profili (Studenti, Genitori, Tutor).
2. **Gestione Lezione:** La parte centrale del sistema, che include ricerca, prenotazione, pagamento e gestione disponibilità.
3. **Recensione:** Sistema di feedback post-lezione.
4. **Segnalazione:** Modulo di moderazione per utenti con comportamenti scorretti.
5. **Persistenza:** Interazione con il database MySQL tramite pattern DAO.

4. Features to be tested/not to be tested

Le seguenti funzionalità, derivate dalla scomposizione dei sottosistemi, saranno oggetto di test:

Sottosistema	Funzionalità	Priorità
<i>Gestione Utente</i>	Login, Registrazione (con distinzione ruoli).	Alta

<i>Gestione Utenza</i>	Modifica Profilo	Media
<i>Gestione Lezione</i>	Ricerca Lezioni, Prenotazione (check disponibilità), Annullamento.	Alta
<i>Gestione Lezione</i>	Modifica informazioni lezione (Tutor)	Media
<i>Recensioni</i>	Inserimento Recensione, Visualizzazione media voti.	Bassa
<i>Segnalazioni</i>	Invio segnalazione, Moderazione (Ban utente).	Media

Tutte le funzionalità con priorità medio/bassa non verranno testate, nello specifico anche:

- Gateway di Pagamento Reale: L'interazione con i circuiti bancari reali non sarà testata; verrà utilizzato un mock per simulare l'esito del pagamento.
- Servizio Email esterno: L'invio effettivo delle email sarà simulato tramite log di sistema o mock object per evitare spam durante il testing.

5. Pass/Fail criteria

I criteri per determinare il successo o il fallimento di un test sono i seguenti:

- **Pass (Successo):** Il risultato ottenuto (Output reale) coincide con il risultato atteso (Oracolo). Inoltre, non vengono sollevate eccezioni non gestite e lo stato del database riflette le modifiche previste.
- **Fail (Fallimento):** Il risultato ottenuto differisce da quello atteso, il sistema va in crash, o viene mostrata una pagina di errore generica (HTTP 500) invece di un messaggio di errore controllato.

6. Approach

La strategia di testing segue un approccio incrementale suddiviso in tre livelli:

- Testing di Unità
- Testing di Integrazione
- Testing di Sistema

6.1 Testing di Unità (White-Box)

Questa fase si concentra sulla verifica delle singole componenti software (classi), isolandole dalle loro dipendenze.

- Oggetto: Classi del livello Storage (DAO).

- Strategia: White-box testing mirato alla copertura dei rami condizionali (Branch Coverage).
- Tool: Framework JUnit 5.

6.2 Testing di Integrazione

Verifica la corretta interazione tra i sottosistemi e i livelli architetturali (Controller \longleftrightarrow Manager \longleftrightarrow DAO).

- Strategia: Approccio Bottom-up, integrando progressivamente i componenti dai livelli più bassi (Storage) a quelli più alti (Presentation).
- Tool: Mockito per simulare le dipendenze esterne durante i test.

6.3 Testing di Sistema (Black-Box)

Valida il sistema completo simulando il comportamento dell'utente finale per accertare il soddisfacimento dei requisiti funzionali e non funzionali.

- Oggetto: Interfaccia Web e flussi utente completi (es. Prenotazione Lezione).
- Strategia: Black-box testing basato sugli scenari d'uso del RAD.
- Tool: Selenium WebDriver per l'automazione dell'interazione browser.

7. Suspension and resumption

In questa sezione verranno illustrati i criteri di sospensione del test e le attività di test che dovranno essere ripetute quando si riprende il test.

- Criteri di Sospensione: Il testing verrà sospeso se viene individuato un “Blocking Bug” (es. impossibilità di effettuare il login o crash del database) che impedisce l'esecuzione dei test successivi.
- Criteri di Ripresa: Il testing verrà ripreso solo quando tutti i problemi relativi alla sospensione dello stesso sono stati risolti. L'attività di testing riprenderà dal test case che ha causato la sospensione.

8. Testing materials (hardware/software requirements)

- Hardware: Workstation di sviluppo (Windows/Mac) con almeno 8GB RAM.
- Ambiente Software: Java Development Kit (JDK) 23, Apache Tomcat 10.1, MySQL Server 8.0.
- Software Testing:
 - Eclipse IDE / IntelliJ IDEA.
 - JUnit 5 (Framework di test).
 - Mockito (Framework di mocking).
 - Selenium WebDriver (Browser automation).
- Dati di test: Un database popolato con dati fittizi (Studenti, Tutor, Lezioni pre-inserite).

9. Test cases

9.1 Testing di Sistema (Black-Box)

Obiettivo: Verificare che il sistema soddisfi i Requisiti Funzionali descritti nel RAD senza preoccuparsi del codice interno. Questo livello di test si concentra sugli input (dati inseriti dallo studente) e sugli output (successo prenotazione o messaggio di errore).

- Tecnica: Equivalence Partitioning (Partizione delle classi di equivalenza) e Boundary Value Analysis (Analisi dei valori limite).

Educat	Ingegneria del Software	Pagina 6 di 21
--------	-------------------------	----------------

- Componenti sotto test: Interfaccia Utente (GestioneLezioneGUI) verso il Controller (PrenotazioneServlet).
- Scenari di Test (Black Box):

Test Case	Input	Risultato Atteso
<i>TC_LEZ_01: Prenotazione con successo (Happy Path)</i>	Utente “Studente” autenticato, ID Lezione valida, Slot orario libero, Dati carta di credito validi (formato corretto).	Il sistema restituisce la conferma, invia l'email e lo stato della prenotazione diventa “attiva”.
<i>TC_LEZ_02: Nuova lezione tutor</i>	Utente “Tutor” autenticato, pagina di “Nuova lezione”.	Il sistema inserisce la disponibilità nel database, imposta la lezione come “Pianificata”, reindirizza il tutor alla pagina delle lezioni.
<i>TC LEZ_03: Annullamento lezione tutor</i>	Utente “Tutor” autenticato, pagina delle prenotazioni aperta, prenotazione effettuata da uno Studente per una delle lezioni offerte dal Tutor.	Il sistema imposta la prenotazione dello studente come “Annullata” e la lezione torna nuovamente disponibile “Pianificata”.
<i>TC LEZ_04: Annullamento lezione studente</i>	Utente “Studente” autenticato, pagina delle prenotazioni aperta, prenotazione effettuata da uno Studente.	Il sistema imposta la prenotazione dello studente come “Annullata” e la lezione torna nuovamente disponibile “Pianificata”.
<i>TC LEZ_05: Login</i>	L’utente non autenticato si trova sulla pagina del login.	L’utente è autenticato e viene indirizzato alla pagina home relativa al proprio ruolo.
<i>TC LEZ_06: Registrazione</i>	L’utente si trova sulla pagine home generica del sito	Le informazioni del nuovo utente vengono salvate nel database e viene reindirizzato alla pagina home corrispondente al ruolo scelto.
<i>TC LEZ_07: Ricerca lezioni</i>	L’utente autenticato con ruolo “Studente” o “Genitore” si trova sulla pagina della ricerca delle lezioni.	Il sistema mostra all’utente la lista delle lezioni che rispettano i parametri.

9.2. Applicazione Tecnica Category Partition

9.2.1. TC_LEZ_01 Prenotazione

ID Lezione:

ID1: empty	[error]
ID2: ID non corretto	[error]
ID3: ID corretto	[if NonEmpty] [property Match]

Data Lezione:

DL1: Inizio lezione < current Time +1 giorno	[error]
DL2: Inizio lezione > current Time	[if Match]
DL3: Fine lezione < current Time	[error]

Slot Lezione:

TS1: Time Slot Libero	[if Match]
TS2: Time Slot Occupato	[error]

Dati di pagamento

Numero carta:

NC1: empty	[error]
NC2: Numero Cifre = 16	[property CifreCartaCorrette]
NC3: Numero Cifre > 0 e ≠ 16	[error]

Codice di sicurezza:

CS1: empty	[error]
CS2: Numero Cifre = 3	[property CifreCVVCorrette]
CS3: Numero Cifre > 0 e ≠ 3	[error]

Data di scadenza:

DS1: empty	[error]
DS2: Data di scadenza < current Date	[error]
DS3: Data di scadenza > current Date	[property DataValida]

Titolare della carta:

TC1: empty	[error]
TC2: Nome/Cognome contengono numeri e/o caratteri speciali	[error]
TC3: Nome/Cognome corrispondono al Numero Carta	[if NumeroCartaCorretto] [property TitolareCartaCorretto]

DP1: empty	[error]
DP2: Dati di pagamento corretti	[if CifreCartaCorrette] [if CifreCVVCorrette] [if DataScadenzaCorretta] [if TitolareCartaCorretto]
DP3: Dati di pagamento non corretti	[error]

Test Frame

TC1: ID3, DL3, TS1, NC2, CS2, DS3, TC3, DP2 – ORACOLO: Prenotazione effettuata, utente viene reindirizzato alla pagina delle prenotazioni con un messaggio di conferma.

TC2: ID1, DL3, TS1, NC2, CS2, DS3, TC3, DP2 – ORACOLO: Utente viene reindirizzato alla pagina della ricerca lezioni con un messaggio di errore “Lezione non trovata”.

TC3: ID2, DL3, TS1, NC2, CS2, DS3, TC3, DP2 – ORACOLO: Utente viene reindirizzato alla pagina delle prenotazioni con un messaggio di errore “Errore nella prenotazione”.

TC4: ID3, DL1, TS1, NC2, CS2, DS3, TC3, DP2 – ORACOLO: Utente viene reindirizzato alla pagina delle prenotazioni con un messaggio di errore “Non puoi annullare la prenotazione entro un giorno dall’inizio della lezione”.

TC5: ID3, DL3, TS1, NC2, CS2, DS3, TC3, DP2 – ORACOLO: Utente viene reindirizzato alla pagina delle prenotazioni con un messaggio di errore “La data di questa lezione è passata.”.

TC6: ID3, DL2, TS2, NC2, CS2, DS3, TC3, DP2 – ORACOLO: Utente viene reindirizzato alla pagina delle prenotazioni con un messaggio di errore “Hai già una prenotazione attiva per in questo slot temporale”.

TC7: ID3, DL2, TS1, NC1, CS2, DS3, TC3, DP2 – ORACOLO: Utente viene reindirizzato alla pagina delle prenotazioni con un messaggio di errore “Inserire un numero di carta”.

TC8: ID3, DL2, TS1, NC3, CS2, DS3, TC3, DP2 – ORACOLO: Utente viene reindirizzato alla pagina delle prenotazioni con un messaggio di errore “Il numero di carta deve contenere 16 cifre”.

TC9: ID3, DL3, TS1, NC2, CS1, DS3, TC3, DP2 – ORACOLO: Utente viene reindirizzato alla pagina delle prenotazioni con un messaggio di errore “Inserire un CVV”.

TC10: ID3, DL3, TS1, NC2, CS4, DS3, TC3, DP2 – ORACOLO: Utente viene reindirizzato alla pagina delle prenotazioni con un messaggio di errore “Il CVV deve contenere 3 cifre”.

TC11: ID3, DL3, TS1, NC2, CS2, DS1, TC3, DP2 – ORACOLO: Utente viene reindirizzato alla pagina delle prenotazioni con un messaggio di errore “Inserire data di scadenza”.

TC12: ID3, DL3, TS1, NC2, CS2, DS2, TC3, DP2 – ORACOLO: Utente viene reindirizzato alla pagina delle prenotazioni con un messaggio di errore “Carta scaduta”.

TC13: ID3, DL3, TS1, NC2, CS2, DS3, TC1, DP2 – ORACOLO: Utente viene reindirizzato alla pagina delle prenotazioni con un messaggio di errore “Inserire titolare carta”.

TC14: ID3, DL3, TS1, NC2, CS2, DS3, TC2, DP2 – ORACOLO: Utente viene reindirizzato alla pagina delle prenotazioni con un messaggio di errore “Il titolare della carta non può contenere numeri e/o caratteri speciali”.

TC15: ID3, DL3, TS1, NC2, CS2, DS3, TC3, DP1 – ORACOLO: Utente viene reindirizzato alla pagina delle prenotazioni con un messaggio di errore “Inserire dati di pagamento”.

TC16: ID3, DL3, TS1, NC2, CS2, DS3, TC3, DP3 – ORACOLO: Utente viene reindirizzato alla pagina delle prenotazioni con un messaggio di errore “Dati di pagamento non corretti”.

9.2.2. TC_LEZ_02 Nuova lezione tutor

Materia

M1: empty	[error]
M2: Selezionata una delle opzioni	[property MateriaSelezionata]

Giorno della lezione

GL1: empty	[error]
GL2: Selezionata una data	[property DataSelezionata]
GL3: Data odierna > Data selezionata	[error]

Ora della lezione

OL1: empty	[error]
OL2: Selezionata solo ora inizio	[error]
OL3: Selezionata solo ora fine	[error]
OL4: Ora fine < Ora inizio	[error]
OL5: Ora fine > Ora inizio	[property OraCorretta]

Prezzo orario

PO1: empty	[error]
PO2: Selezionato prezzo	[property PrezzoScelto]

Modalità

MO1: empty	[error]
-------------------	---------

MO2: Selezionata opzione	[property ModalitàScelta]
---------------------------------	---------------------------

Slot temporale

ST1: Lezione non va in conflitto con una già offerta	[property NonConflitto]
ST2: Lezione va in conflitto con una già scelta	[error]

Test Frame

TC1: M2, GL2, OL5, PO2, MO2, ST1 – ORACOLO: Il sistema inserisce la disponibilità nel database, imposta la lezione come “Pianificata”, reindirizza il tutor alla pagina delle lezioni.

TC2: M1, GL2, OL5, PO2, MO2, ST1 – ORACOLO: Il sistema riporta l’utente sulla pagina della nuova lezione con un messaggio di errore “Selezionare una materia”.

TC3: M2, GL1, OL5, PO2, MO2, ST1 – ORACOLO: Il sistema riporta l’utente sulla pagina della nuova lezione con un messaggio di errore “Selezionare una data”.

TC4: M2, GL3, OL5, PO2, MO2, ST1 – ORACOLO: Il sistema riporta l’utente sulla pagina della nuova lezione con un messaggio di errore “Impossibile selezionare una data passata”.

TC5: M2, GL2, OL1, PO2, MO2, ST1 – ORACOLO: Il sistema riporta l’utente sulla pagina della nuova lezione con un messaggio di errore “Selezionare un orario”.

TC6: M2, GL2, OL2, PO2, MO2, ST1 – ORACOLO: Il sistema riporta l’utente sulla pagina della nuova lezione con un messaggio di errore “Selezionare ora fine lezione”.

TC7: M2, GL2, OL3, PO2, MO2, ST1 – ORACOLO: Il sistema riporta l’utente sulla pagina della nuova lezione con un messaggio di errore “Selezionare ora inizio lezione”.

TC8: M2, GL2, OL4, PO2, MO2, ST1 – ORACOLO: Il sistema riporta l’utente sulla pagina della nuova lezione con un messaggio di errore “L’ora di fine lezione non può essere inferiore all’ora di inizio”.

TC9: M2, GL2, OL5, PO1, MO2, ST1 – ORACOLO: Il sistema riporta l’utente sulla pagina della nuova lezione con un messaggio di errore “Selezionare un prezzo orario”.

TC10: M2, GL2, OL5, PO2, MO1, ST1 – ORACOLO: Il sistema riporta l’utente sulla pagina della nuova lezione con un messaggio di errore “Selezionare modalità”.

TC11: M2, GL2, OL5, PO2, MO1, ST2 – ORACOLO: Il sistema riporta l’utente sulla pagina della nuova lezione con un messaggio di errore “Hai già una lezione pianificata per questa fascia oraria”.

9.2.3. TC_LEZ_03 Annullamento lezione (Tutor)

ID Prenotazione:

ID1: empty	[error]
ID2: ID corretto	[property IDCorretto]

Tempo rimanente all’inizio della lezione:

TR1: Inizio lezione < current Time +1 giorno	[error]
---	---------

TR2: Inizio lezione > current Time +1 giorno	[if IDCorretto]
TR3: mancano 23h e 59m	[error]
TR4: mancano 24h 1m	[if IDCorretto]

Test Frame

TC1: ID2, TR2 – ORACOLO: Il sistema porta l’utente alla pagina delle prenotazioni con un messaggio “Prenotazione annullata”. Il sistema imposta la prenotazione dello studente come “Annullata” e la lezione torna nuovamente disponibile “Pianificata”.

TC2: ID1, TR2 – ORACOLO: Il sistema porta l’utente alla pagina delle prenotazioni con un messaggio di errore “ID Prenotazione errato”.

TC3: ID2, TR1 – ORACOLO: Il sistema porta l’utente alla pagina delle prenotazioni con un messaggio di errore “Impossibile eliminare la prenotazione entro un giorno dall’inizio della lezione”.

TC4: ID2, TR3 – ORACOLO: Il sistema porta l’utente alla pagina delle prenotazioni con un messaggio di errore “Impossibile eliminare la prenotazione entro un giorno dall’inizio della lezione”.

TC5: ID2, TR4 – ORACOLO: Il sistema porta l’utente alla pagina delle prenotazioni con un messaggio “Prenotazione annullata”. Il sistema imposta la prenotazione dello studente come “Annullata” e la lezione torna nuovamente disponibile “Pianificata”.

9.2.4. TC_LEZ_04 Annullamento lezione (Studente)

ID Prenotazione:

ID1: empty	[error]
ID2: ID corretto	[property IDCorretto]

Tempo rimanente all’inizio della lezione:

TR1: Inizio lezione < current Time +1 giorno	[error]
TR2: Inizio lezione > current Time +1 giorno	[if IDCorretto]
TR3: mancano 23h e 59m	[error]
TR4: mancano 24h 1m	[if IDCorretto]

Test Frame

TC1: ID2, TR2 – ORACOLO: Il sistema porta l’utente alla pagina delle prenotazioni con un messaggio “Prenotazione annullata”. Il sistema imposta la prenotazione dello studente come

“Annullata” e la lezione torna nuovamente disponibile “Pianificata”.

TC2: ID1, TR2 – ORACOLO: Il sistema porta l’utente alla pagina delle prenotazioni con un messaggio di errore “ID Prenotazione errato”.

TC3: ID2, TR1 – ORACOLO: Il sistema porta l’utente alla pagina delle prenotazioni con un messaggio di errore “Impossibile eliminare la prenotazione entro un giorno dall’inizio della lezione”.

TC4: ID2, TR3 – ORACOLO: Il sistema porta l’utente alla pagina delle prenotazioni con un messaggio di errore “Impossibile eliminare la prenotazione entro un giorno dall’inizio della lezione”.

TC5: ID2, TR4 – ORACOLO: Il sistema porta l’utente alla pagina delle prenotazioni con un messaggio “Prenotazione annullata”. Il sistema imposta la prenotazione dello studente come “Annullata” e la lezione torna nuovamente disponibile “Pianificata”.

9.2.5. TC_LEZ_05 Login

Mail:

UP1: mail presente nel database	[property UsernamePresente]
UP2: mail NON presente nel database	[error]
UP3: empty	[error]

Password:

PW1: password associata a mail nel database	[if UsernamePresente]
PW2: password NON associata a mail nel database	[error]
PW3: empty	[error]

Test Frame

TC1: UP1, PW1 – ORACOLO: L’utente è autenticato e viene indirizzato alla pagina home relativa al proprio ruolo.

TC2: UP2, PW1 – ORACOLO: Il sistema porta di nuovo l’utente alla pagina del login con un messaggio di errore “Credenziali errate”.

TC3: UP3, PW1 – ORACOLO: Il sistema porta di nuovo l’utente alla pagina del login con un messaggio di errore “Inserire email”.

TC4: UP1, PW2 – ORACOLO: Il sistema porta di nuovo l’utente alla pagina del login con un messaggio di errore “Email o password errata”.

TC5: UP1, PW3 – ORACOLO: Il sistema porta di nuovo l’utente alla pagina del login con un messaggio di errore “Inserire password”.

9.2.6. TC_LEZ_06 Registrazione

Ruolo:

R1: selezionato ruolo “Studente”	
----------------------------------	--

R2: selezionato ruolo “Genitore”	[property RuoloGenitore]
R3: selezionato ruolo “Tutor”	

Nome e Cognome:

NC1: nome e cognome di formato valido	
NC2: nome e cognome di formato NON valido	[error]

Email:

E1: email di formato valido	
E2: email di formato NON valido	[error]

Indirizzo:

I1: indirizzo di formato valido	
I2: indirizzo di formato NON valido	[error]

Data di Nascita:

DN1: data selezionata valida	
DN2: l’utente ha meno di 18 anni	[error]

Dati figlio:

DF1: nome e cognome di formato valido	[if RuoloGenitore]
DF2: nome e cognome di formato NON valido	[error]

Data di nascita figlio:

DNF1: data selezionata valida	[if RuoloGenitore]
DNF2: il figlio è più grande del genitore	[error]

Test Frame

TC1: R1, NC1, E1, I1, DN1 – ORACOLO: Le informazioni del nuovo utente vengono salvate nel database e viene reindirizzato alla pagina home corrispondente al ruolo scelto.

TC2: R2, NC1, E1, I1, DN1, DF1, DNF1 – ORACOLO: Le informazioni del nuovo utente vengono salvate nel database e viene reindirizzato alla pagina home corrispondente al ruolo scelto.

TC3: R2, NC1, E1, I1, DN1, DF2, DNF1 – ORACOLO: L’utente viene riportato alla pagina della registrazione col messaggio di errore “Nome/Cognome di formato non valido”.

TC4: R2, NC1, E1, I1, DN1, DF1, DNF2 – ORACOLO: L’utente viene riportato alla pagina della

registrazione col messaggio di errore “Il figlio/la figlia non può essere più grande del genitore”.

TC5: R1, NC2, E1, I1, DN1 – ORACOLO: L’utente viene riportato alla pagina della registrazione col messaggio di errore “Nome/Cognome di formato non valido”.

TC6: R1, NC1, E2, I1, DN1 – ORACOLO: L’utente viene riportato alla pagina della registrazione col messaggio di errore “Email di formato non valido”.

TC7: R1, NC1, E2, I2, DN1 – ORACOLO: L’utente viene riportato alla pagina della registrazione col messaggio di errore “Indirizzo di formato non valido”.

TC8: R1, NC1, E2, I2, DN2 – ORACOLO: L’utente viene riportato alla pagina della registrazione col messaggio di errore “Lo studente deve essere maggiorenne”.

9.2.7. TC_LEZ_07 Ricerca lezione

Materia:

M1: materia selezionata	[single]
M2: materia non selezionata	

Modalità:

MO1: modalità selezionata	[single]
MO2: modalità non selezionata	

Città:

C1: città selezionata	[single]
C2: città non selezionata	

Prezzo:

P1: prezzo massimo inserito	[single]
P2: prezzo massimo non inserito	

Test Frame

TC1: M2, MO2, C2, P2 – ORACOLO: Il sistema mostra tutte le lezioni prenotabili.

TC2: M1, MO2, C2, P2 – ORACOLO: Il sistema mostra le lezioni filtrate secondo i criteri richiesti.

TC3: M2, MO1, C2, P2 – ORACOLO: Il sistema mostra le lezioni filtrate secondo i criteri richiesti.

TC4: M2, MO2, C1, P2 – ORACOLO: Il sistema mostra le lezioni filtrate secondo i criteri richiesti.

TC5: M2, MO2, C2, P1 – ORACOLO: Il sistema mostra le lezioni filtrate secondo i criteri richiesti.

9.3 Testing Strutturale (White-Box)

- Obiettivo: Analisi del codice interno, in particolare la logica di “gestionelezione” definita nell’ODD, per assicurarci che tutti i percorsi (branch coverage) e le gestioni delle eccezioni siano testati.
- Tecnica: Branch Coverage e Path Testing.

- Componente sotto test: “it.unisa.educat.controller.gestionelezione”, : “it.unisa.educat.controller.gestioneutenza” e le sue interazioni con il DAO.
- Scenari di Test (White Box):

Test Case	Logica da testare
<i>WB_TC_01: Inserimento Lezione (doSaveLezione)</i>	Metodo GestioneLezioneDAO.doSaveLezione()
<i>WB_TC_02:Prenotazione Lezione (prenotaLezione)</i>	Metodo GestioneLezioneDAO.prenotaLezione()
<i>WB_TC_03:Annullamento Prenotazione (annullaPrenotazione)</i>	Metodo GestioneLezioneDAO.annullaPrenotazione()
<i>WB_TC_04: Fascia oraria lezione (Tutor)</i>	Metodo GestioneLezioneDAO.hasTutorLezioneInFasciaOraria()
<i>WB_TC_05: Ricerca con criteri</i>	Metodo GestioneLezioneDAO.doRetrieveByCriteria()
<i>WB_TC_06: Mapping ResultSet</i>	Metodo GestioneLezioneDAO.mapResultSetToLezione()
<i>WB_TC_07: Inserimento Utente</i>	Metodo GestioneUtenzaDAO.doSave()
<i>WB_TC_08: Recupero per Email</i>	Metodo GestioneUtenzaDAO.doRetrieveByEmail()
<i>WB_TC_09: Recupero per ID</i>	Metodo GestioneUtenzaDAO.doRetrieveByID()
<i>WB_TC_10: Recupero tutti Utenti</i>	Metodo GestioneUtenzaDAO.doRetrieveAll()
<i>WB_TC_11: Aggiornamento Utente</i>	Metodo GestioneUtenzaDAO.doUpdate()
<i>WB_TC_12: Eliminazione Utente</i>	Metodo GestioneUtenzaDAO.doDelete()
<i>WB_TC_13: Mapping ResultSet a Utente</i>	Metodo GestioneUtenzaDAO.mapResultSetToUtente()

9.3.1 WB_TC_01 Nuova lezione tutor

Execution Result:

ER1: executeUpdate() restituisce 1	
ER2: executeUpdate() restituisce 0	[error]
ER3: executeUpdate() lancia SQLException	[error]

Test Frame

- TC1:** ER1 – ORACOLO: true, lezione salvata con ID
TC2: ER2 – ORACOLO: false, nessuna riga inserita.
TC3: ER3 – ORACOLO: false, SQLException propagata.

9.3.2 WB_TC_02 Prenotazione lezione

Execution Result:

ER1: executeUpdate() restituisce 1	
ER2: executeUpdate() restituisce 0	[error]
ER3: executeUpdate() lancia SQLException	[error]

Test Frame

- TC1:** ER1 – ORACOLO: true, prenotazione salvata.
TC2: ER2 – ORACOLO: false, nessuna riga inserita.
TC3: ER3 – ORACOLO: false, SQLException propagata.

9.3.3 WB_TC_03 Annullamento Prenotazione

Update Prenotazione:

UP1: executeUpdate() restituisce 1	[property PrenotazioneAggiornata]
UP2: executeUpdate() restituisce 0	[error]
UP3: executeUpdate() lancia SQLException	[error]

Update Lezione:

UL1: executeUpdate() restituisce 1	[if PrenotazioneAggiornata]
UL2: executeUpdate() restituisce 0	[error]
UL3: executeUpdate() lancia SQLException	[error]

Test Frame

- TC1:** UP1, UL1 – ORACOLO: true, transazione completata
TC2: UP2, UL1 – ORACOLO: false, rollback.
TC2: UP1, UL2 – ORACOLO: false, rollback.
TC3: UP3, UL1 – ORACOLO: SQLException, rollback.

9.3.4 WB_TC_04 Fascia oraria lezione (Tutor)

Risultato Query:

RQ1: COUNT = 0	
RQ2: COUNT > 0	
RQ3: lancia SQLException	[error]

Test Frame**TC1:** RQ1 – ORACOLO: false.**TC2:** RQ2 – ORACOLO: true.**TC3:** RQ3 – ORACOLO: SQLException.**9.3.5 WB_TC_05 Ricerca con Criteri****Criteri:**

CR1: empty	
CR2: Filtri inseriti	
CR3: Per idTutor	

Test Frame**TC1:** CR1 – ORACOLO: lista vuota.**TC2:** CR2 – ORACOLO: lista filtrate.**TC3:** CR3 – ORACOLO: lista per tutor.**9.3.6 WB_TC_06 Mapping ResultSet****Stati Lezione:**

SL1: “PIANIFICATA”	[single]
SL2: “PRENOTATA”	[single]

Modalità Lezione:

ML1: “ONLINE”	[single]
ML2: “PRESENZA”	[single]

Test Frame**TC1:** SL1, ML1 – ORACOLO: lezione mappata correttamente.**TC2:** SL2, ML2 – ORACOLO: lezione mappata correttamente.**9.3.7 WB_TC_07 Inserimento Utente****Tipo Utente:**

SL1: “STUDENTE”	
SL2: “GENITORE”	[single]
SL3: “TUTOR”	[single]

Esito:

EO1: riga inserita e ID generato	
EO2: nessuna riga inserita	[error]
EO3: lancia SQLException	[error]

Test Frame

TC1: SL1 + EO1 – ORACOLO: true, studente salvato con ID.

TC2: SL2 + EO1 – ORACOLO: true, genitore salvato con ID.

TC3: SL3 + EO1 – ORACOLO: true, tutor salvato con ID.

TC4: SL1 + EO2 – ORACOLO: false, nessuna riga inserita.

TC5: SL1 + EO3 – ORACOLO: false, eccezione propagata.

9.3.8 WB_TC_08 Recupero per Email

Email:

E1: Email presente nel DB	[property EmailPresente]
E2: Email non presente nel DB	[property EmailNonPresente]

Esito:

ES1: positivo	[if EmailPresente]
ES2: negativo	[if EmailNonPresente]
ES3: lancia SQLException	[error]

Test Frame

TC1: E1 + ES1 – ORACOLO: true, UtenteDTO mappato correttamente.

TC2: E2 + ES2 – ORACOLO: null, email non trovata.

TC3: E1 + ES3 – ORACOLO: SQLException propagata.

9.3.9 WB_TC_09 Recupero per ID

ID:

ID1: ID presente nel DB	[property IDPresente]
ID2: Email non presente nel DB	[property IDNonPresente]

Esito:

ES1: positivo	[if IDPresente]
ES2: negativo	[if IDNonPresente]
ES3: lancia SQLException	[error]

Test Frame

TC1: ID1 + ES1 – ORACOLO: true, UtenteDTO mappato correttamente.

TC2: ID2 + ES1 – ORACOLO: null, email non trovata.

TC3: ID1 + ES3 – ORACOLO: SQLException propagata.

9.3.10 WB_TC_10 Recupero tutti Utenti**Database:**

DB1: Il database è vuoto	
DB2: Nel database sono presenti dati	
DB3: lancia SQLException	[error]

Test Frame

TC1: DB1 – ORACOLO: true, UtenteDTO mappato correttamente.

TC2: DB2 – ORACOLO: null.

TC3: DB3 – ORACOLO: SQLException propagata.

9.3.11 WB_TC_11 Aggiornamento Utente**ID:**

ID1: ID utente esistente	[property IDPresente]
ID2: ID utente non esistente	[property IDNonPresente]

Esito:

EU1: executeUpdate() restituisce 1	[if IDPresente]
EU2: executeUpdate() restituisce 0	[if IDNonPresente]
EU3: lancia SQLException	[error]

Test Frame

TC1: ID1 + EU1 – ORACOLO: true, utente aggiornato correttamente.

TC2: ID2 + EU2 – ORACOLO: false, nessuna riga aggiornata.

TC3: ID1 + EU3 – ORACOLO: SQLException propagata

9.3.12 WB_TC_12 Eliminazione Utente**ID:**

ID1: ID utente esistente	[property IDPresente]
ID2: ID utente non esistente	[property IDNonPresente]

Esito:

EU1: executeUpdate() restituisce 1	[if IDPresente]
EU2: executeUpdate() restituisce 0	[if IDNonPresente]
EU3: lancia SQLException	[error]

Test Frame

TC1: ID1 + EU1 – ORACOLO: true, utente eliminato correttamente.

TC2: ID2 + EU2 – ORACOLO: false, nessun utente eliminato.

TC3: ID1 + EU3 – ORACOLO: SQLException propagata.

9.3.13 WB_TC_13 Mapping ResultSet a Utente

Tipo:

T1: Tipo utente valido	
T2: Tipo utente sconosciuto	[error]

Test Frame

TC1: T1 – ORACOLO: true, UtenteDTO mappato correttamente.

TC2: T2 – ORACOLO: false, utente non mappato.