

# TESTING lesson-dn-brognoli-colombo

---

## Sommario

IDENTIFICAZIONE DEGLI ATTORI .....	1
SCENARI D'USO.....	2
CLASS DIAGRAM .....	2
ALLOY.....	2
DESIGN PATTERN.....	3
MODELLO ER.....	3
SESSION BEAN .....	3
DESIGN.....	3

## RASD

---

### IDENTIFICAZIONE DEGLI ATTORI

Nel documento RASD vengono elencate le funzionalità che ogni tipologia di utente, interagendo con il sistema può effettuare:

#### TRAINER:

- Crea corsi e ne determina la categoria
- Crea learning object
- Crea test di valutazione
- Processa le richieste di iscrizione
- Corregge test e ne assegna i voti

#### TRAINEE

- Si iscrive ai corsi offerti
- Esegue test
- Effettua download

#### ASSISTANT

- Corregge i test e assegna i voti

Tutte le richieste date dal documento iniziale sono qui citate quindi riprendono il quadro generale del sistema.

## SCENARI D'USO

Lo use case diagram rappresentato è completo e vengono correttamente gestite le interazioni tra i vari utenti e in particolare non si notano discrepanze tra le funzionalità sopra descritte e quanto analizzato dal grafico.

Vengono poi analizzati alcuni casi d'uso:

correzione test: viene analizzato come il trainer interagisca con il sistema per la correzione di un test

a livello sw verranno utilizzati valori non usuali per vedere il comportamento del sistema (se gli errori sono gestiti correttamente) inserendo per esempio user e pass nulle oppure non esistenti o ancora tecniche semplici di sql injection come 1=1; poi simulando il comportamento di un trainer inseriremo voti composti da stringhe non numeriche.

In questo caso ci si aspetta che il sistema richieda di inserire valori corretti.

Creazione di test: anche in questo caso viene analizzato come il trainer possa creare un test

Verrà seguito passo passo quanto riportato nell'activity diagram verificando che non ci siano situazioni anomale (anche inserendo valori e parametri non idonei)

Esecuzione test: caso in cui il trainee effettua il test

L'utente dovrà accedere correttamente al sistema inoltre si verificheranno che le eccezioni siano implementate in modo da non riscontrare errori nel sistema ovvero:

- Che il test non sia stato ricevuto dallo studente
- Che il test non sia stato ricevuto dal sistema

Iscrizione al corso: ultimo caso d'uso che vede protagonisti l'insegnante e lo studente

Si verificherà che la procedura di iscrizione vada a buon fine. **Va fatto notare un errore nella citazione della condizione d'uscita in cui è analizzata la stessa situazione del caso precedente.**

## CLASS DIAGRAM

Vengono analizzate tutte le classi e a livello sw; si analizzerà se queste saranno implementate ed inoltre se ci saranno differenze con la documentazione DD ( nel qual caso dovranno essere mostrate le rettifiche prima dello sviluppo del sistema).

## ALLOY

Questa ultima parte riprende il linguaggio naturale e lo formalizza eliminando le ambiguità

Nello specifico vengono analizzati tutti i vincoli e non si riscontrano delle mancanze.

Il codice riprende tali vincoli con estrema correttezza e il mondo mostrato è perfettamente coerente con quanto implementato in alloy e quanto citato nel testo.

# DD

---

Il documento presenta la struttura multi tier utilizzata e coerente con quanto indicato all'inizio del progetto.

In particolare in questo progetto si utilizza una struttura a tre strati e si analizzano i possibili vantaggi derivati da essa.

## DESIGN PATTERN

Viene poi analizzato il design pattern; viene fatta una descrizione completa e intuitiva precisando come tale strutturazione migliori la comprensione del modello.

## MODELLO ER

Il modello mostrato riprende in modo preciso quanto espresso nel Rasd anche se va precisato che il template diventa una chiave di test senza avere una entità propria contrariamente da quanto espresso nel documento LESSON.

Ci aspettiamo comunque che tali relazioni e la strutturazione delle entità siano riprese in modo fedele nel sw, anche se viene detto che il lato implementativo varierà.

## SESSION BEAN

Viene visualizzato il session bean mostrando i vari manager e le funzioni utilizzate; viene inoltre puntualizzato che trainer , trainee e assistant devono essere statefull

## DESIGN

Questa parte è espressa in modo chiaro e vengono visualizzati i grafici dei vari casi d'uso analizzati nel rasd; analizzandoli non si notano grosse differenze tra i due documenti.

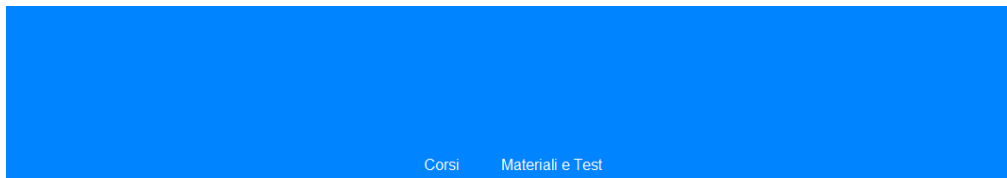
# SOFTWARE

---

Dopo un'analisi fatta sulla documentazione si procede al testing del software.

Vengono qui di seguito riportate delle schermate sui test effettuati.

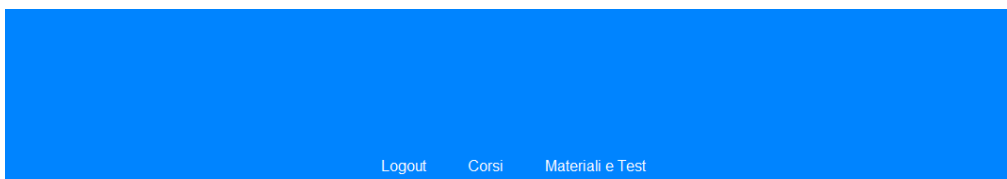
Login con utente di tipo Trainer:



Login avvenuto con successo, **Id:** 1

## Funzionalità Trainer:

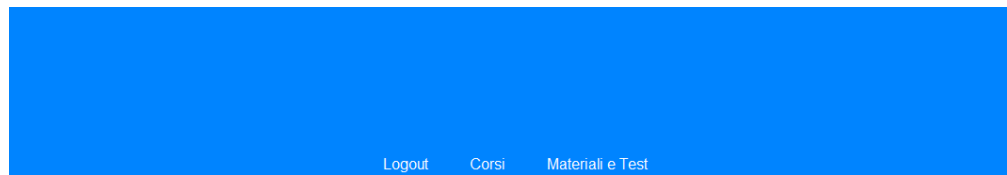
- Creazione corsi



**ID:** 1  
**Nome:** ef  
**Cognome:**  
edwef

Nome corso:   
Topics:   
(inserisci la lista di topic separata da virgole)

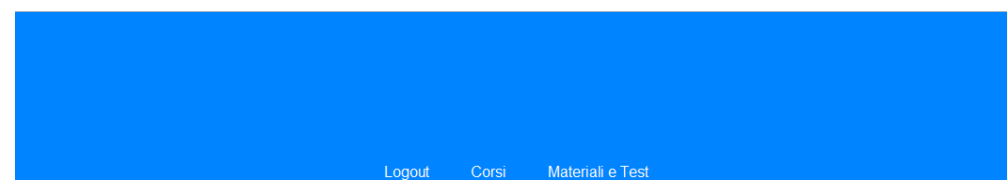
## Aggiunta topics ad un corso:



**ID:** 1  
**Nome:** ef  
**Cognome:**  
edwef

Seleziona il corso:   
Topics:   
(inserisci la lista di topic separata da virgole)

## Un corso può essere associato solo una volta al Trainer:



**ID:** 1  
**Nome:** ef  
**Cognome:**  
edwef



Errore: questo trainer ha già un corso con questo nome

Nome corso:   
Topics:   
(inserisci la lista di topic separata da virgole)

## Associazione assistente – trainer

LogoutCorsiMateriali e Test

ID: 1  
Nome: ef  
Cognome: edwef

Seleziona l'assistente:  
☒ ass ass

Un assistente può essere associato a un solo professore:

LogoutCorsiMateriali e Test


ID: 1  
Nome: ef  
Cognome: edwef

Seleziona l'assistente:  
Non sono presenti assistenti liberi al momento.

Creazione del materiale didattico:

LogoutCorsiMateriali e Test

ID: 1  
Nome: ef  
Cognome: edwef

  
Creato materiale Intro per il corso INGEGNERIA DEL SOFTWARE  
**Per impostare le dipendenze ed i materiali sostitutivi tornare al menu del materiale**

Download materiale didattico:

LogoutCorsiMateriali e Test

ID: 3  
Nome: stu  
Cognome: stu

Seleziona il materiale da scaricare:  
☒ Intro

Creazione test da template

La fase di test non ha mostrato errori tranne 1.

Qualora sia stato effettuato un test l'insegnante/ assistente può correggere infinite volte il compito e quindi inserire di conseguenza infiniti voti.

Analizzandolo potrebbe non essere un errore: il prof potrebbe ricontrollare il compito e rivalutare la situazione, ma tale considerazione non è espressa nelle specifiche iniziali e tanto meno nei documenti analizzati.

A livello di codice invece non si riscontrano differenze sostanziali con quanto espresso nella documentazione formulata.