



Politecnico di Milano

Facoltà di Ingegneria dell'Informazione

Progetto di Ingegneria del Software 2

LESSON

Testing Progetto
lesson-bianchi-castelli

Prof.ssa Di Nitto

A.A. 2010/11

Bonetti Gabriele matr. 766390

Abstract

L'obiettivo di questo documento è di presentare i risultati del testing dell'applicazione fornita, sulla base di un insieme di requisiti minimi fissati.

Il documento è suddiviso in tre sezioni:

- Nella prima sezione viene illustrato l'insieme dei requisiti.
- Nella seconda sezione saranno definiti i casi di test per verificare la conformità dell'applicazione a tali requisiti.
- Infine viene riportato il risultato del comportamento effettivo dell'applicazione e vengono tratte conclusioni qualitative sulla bontà dell'applicazione.

1) Specifica dei requisiti minimi:

Di seguito vengono riportati i requisiti minimi richiesti al fine di garantire l'accettazione dell'applicazione.

1.1) Requisiti generali:

1.1.1) Architetturali:

- a. Il sistema deve essere una applicazione distribuita, basata su thin client (web browser), capace di gestire sessioni simultanee di più client.
- b. L'intero sistema deve essere sviluppato in linguaggio java, su piattaforma JEE.

1.1.2) Cognitivi:

- a. Il sistema deve essere installabile e configurabile da un generico tecnico informatico. Non devono essere richieste conoscenze che vadano oltre la ordinaria amministrazione di sistema.
- b. L'applicazione deve essere utilizzabile da utenti qualsiasi. Non devono essere richieste capacita' che vadano oltre al normale utilizzo di un sistema desktop.

1.1.3) Sicurezza:

- a. La natura dell'applicazione, che non tratta dati estremamente sensibili, richiede requisiti di sicurezza minimi. Deve quindi essere garantito, almeno, un sistema di autenticazione e di protezione delle pagine da accessi non autorizzati.

1.2)Requisiti funzionali:

1.2.1)Registrazione:

- a. Il sistema deve permettere la registrazione di uno studente a cui deve essere assegnata una matricola.
- b. Non devono poter esistere due utenti differenti con la stessa matricola.

1.2.2)Login e logout:

- a. Il sistema deve permettere l'accesso ai soli utenti registrati con successo.
- b. Il sistema deve comportarsi correttamente, e rifiutare l'accesso nel caso di login o password assenti e di login/password non ben formati.
- c. Deve essere possibile effettuare il logout dall'applicazione. Al successivo login lo stato dell'utente nell'applicazione deve essere coerente con lo stato precedente all'ultimo logout.

1.2.3)Creazione Corso

- a. Un Trainer autenticato deve poter procedere alla creazione di un nuovo corso.
- b. Il docente deve vedere il corso appena creato nella propria homepage.
- c. Tutti gli studenti devono vedere il corso creato nella lista dei corsi disponibili per l'iscrizione.

1.2.4)Aggiunta Materiale ad un corso

- a. Un Trainer, dopo aver selezionato uno dei corsi disponibili, deve poter inserire correttamente un nuovo materiale didattico.
- b. Il materiale didattico deve risultare disponibile nella lista dei materiali del corso, sia per il docente che per tutti gli studenti iscritti.

1.2.5)Creazione di un test per un corso

- a. Un Trainer, dopo aver selezionato uno dei corsi disponibili, deve poter inserire correttamente un nuovo test creando domande di tipo: risposta aperta, esercizio, domanda a risposta multipla.
- b. Il test deve risultare disponibile nella lista dei materiali del corso, sia per il docente che per tutti gli studenti iscritti.

1.2.6)Definizione relazioni fra learning objects

- a. Un Trainer, dopo aver selezionato uno dei Learning Objects di un corso, deve poter scegliere, fra i learning objects presenti, quelli necessari per accedere al Learning Object corrente.
- b. Non deve essere possibile, per un qualunque learning object, inserire se stesso nella lista delle precedenze.
- c. Non deve essere possibile, per un qualunque learning object, inserire nella lista delle precedenze, un learning object che abbia nelle sue precedenze il primo oggetto. Non deve esistere quindi, la possibilità di creare precedenze circolari di qualunque grandezza.
- d. Uno studente non deve poter accedere al learning object finché non siano soddisfatte tutte le sue precedenze.

1.2.7) Assegnamento assistente ad un corso

- a. Un Trainer, dopo aver creato un corso deve poter nominare un assistente del corso.

1.2.8) Visualizzazione risultati dei test del corso

- a. Un Trainer, dopo aver selezionato un corso, deve poter visualizzare in modo aggregato i risultati dei test sostenuti dagli studenti di quel corso.
- b. I risultati visualizzati devono essere coerenti con i voti inseriti.

1.2.9) Iscrizione a corso

- a. Uno studente autenticato deve poter accedere a una lista dei corsi disponibili e richiedere l'iscrizione per uno qualunque di essi.
- b. Lo studente non deve potersi iscrivere più volte allo stesso corso.
- c. Il Trainer del corso deve visualizzare correttamente la richiesta d'iscrizione e deve poter decidere se approvarla o rifiutarla.
- d. L'effetto sul sistema deve essere coerente con la decisione del professore, in caso di conferma lo studente deve visualizzare nella sua home il corso.

1.2.10) Scaricamento Materiale

- a. Uno studente, dopo aver selezionato uno dei corsi a cui è iscritto, deve poter accedere alla lista dei learning objects, e, selezionando un materiale, effettuarne il download.

1.2.11) Sostenimento Test

- a. Uno studente, dopo aver selezionato uno dei corsi a cui è iscritto, deve poter accedere alla lista dei learning objects, e, selezionando un test, visualizzare la lista delle domande e poter inserire le relative risposte.
- b. Il sistema deve impedire allo studente di eseguire più volte lo stesso test.

1.2.12) Visualizzazione Voti Test

- a. Uno studente, dalla propria home, deve poter visualizzare un resoconto dei test svolti con i relativi voti.
- b. I risultati visualizzati devono essere coerenti con i voti effettivamente ricevuti.

2)Definizione casi di test:

Di seguito sono riportati i casi di test stabiliti per valutare la compatibilita' con i requisiti minimi.

Modalita' di test:

I test di seguito elencati sono stati effettuati lanciando l'applicazione e osservando il comportamento della stessa. In contemporanea, se necessario, vengono monitorati i cambiamenti nelle tuple della base di dati così da poter visualizzare il corretto comportamento anche nei casi in cui non ci sia un output direttamente osservabile sull'interfaccia.

2.1)Requisiti generali:

2.1.1)Architetturali:

a. Metodologia:

1. Verra' effettuato il deploy dell'applicazione su un sistema così composto:
 - Sistema operativo windows 7
 - Java runtime environment versione 6
 - Jboss Application Server versione 4.2.3.GA
2. La web application verra' testata con due istanze contemporanee di 2 browser differenti: mozilla firefox versione 3.6.13 e google chrome 8.0.552.237.

Risultato atteso:

1. Deploy effettuato correttamente.
2. Applicazione web correttamente visualizzabile dal browser.
3. Sessioni contemporanee funzionanti.

b. Metodologia:

Analisi statica dei sorgenti del progetto.

Risultato atteso:

1. Il progetto deve essere interamente scritto in linguaggio Java ed utilizzare il framework Java EE.

2.1.2)Cognitivi:

a. Metodologia:

L'applicazione sarà installata e configurata seguendo alla lettera il manuale fornito.

L'installazione di eventuale software di terze parti può essere considerata come abilità di base del tecnico installatore.

Risultato atteso:

Al termine della procedura di installazione e configurazione l'applicazione deve funzionare senza errori.

b. Metodologia:

L'applicazione sarà utilizzata con un normale web browser.

Risultato atteso:

L'applicazione deve consentire un semplice accesso senza procedimenti poco chiari o intuitivi.

2.1.3)Sicurezza:

a. Metodologia:

Saranno richiamate le pagine personali di studenti e docenti e le pagine di amministrazione con funzionalità critiche senza aver effettuato il login.

Risultato atteso:

L'applicazione deve impedire l'accesso a tali pagine rendendo impossibile l'utilizzo di qualunque funzionalità ad un utente non autenticato.

2.2)Requisiti funzionali:

2.2.1)Registrazione:

a. Metodologia:

Verra' effettuata la registrazione di un nuvo utente.

a. Risultato atteso:

Il nuovi utenti devono essere registrati correttamente. Il successo deve essere comunicato all'utente.

b. Metodologia:

Verra' effettuata la registrazione di un ulteriore utente.

b. Risultato atteso:

La base di dati deve essere consistente, i due utenti devono avere id differenti.

2.2.2)Login e logout:

a. Metodologia:

1. Verra' effettuato il login con uno username registrato.
2. Verra' effettuato il login con uno username non registrato.

a. Risultato atteso:

1. Il login deve avere successo. L'utente deve poter accedere alle funzionalita' del sistema.
2. Il login deve fallire. L'utente non deve poter accedere alle funzionalita' del sistema.

b. Metodologia:

Verranno effettuate login con: login vuota, password vuota, entrambe vuote, caratteri speciali.

b. Risultato atteso:

Il login deve fallire e l'applicazione comportarsi in modo robusto.

c. Metodologia:

Login di un utente registrato, iscrizione ad un corso, logout. Quindi nuovo login.

c. Risultato atteso:

Lo stato delle iscrizioni deve rimanere identico alla situazione pre-logout.

2.2.2) Creazione Corso:

a. Metodologia:

Verrà effettuata la creazione di un nuovo corso.

a. Risultato atteso:

La creazione deve avvenire con successo, ciò sarà verificato sulla base di dati.

b. Metodologia:

Verrà effettuata la creazione di un nuovo corso e si tornerà alla home page del Trainer.

b. Risultato atteso:

Il nuovo corso deve essere visualizzato correttamente.

c. Metodologia:

Verrà effettuata la creazione di un nuovo corso, si effettuerà il logout e si accederà come studente alla lista dei corsi disponibili.

c. Risultato atteso:

Il nuovo corso deve essere visualizzato correttamente.

2.2.4) Aggiunta Materiale ad un corso:

a. Metodologia:

Verrà selezionato un corso esistente e si effettuerà la creazione di un nuovo materiale.

a. Risultato atteso:

La creazione deve avvenire con successo.

b .Metodologia:

Verrà selezionato un corso esistente e si effettuerà la creazione di un nuovo materiale. In seguito si eseguirà il logout e si accederà come uno degli utenti iscritti al corso.

b. Risultato atteso:

Il nuovo materiale deve essere visualizzato correttamente nella lista dei learning objects del corso.

2.2.5) Creazione di un test per un corso:

a. Metodologia:

Verrà selezionato un corso esistente e si effettuerà la creazione di un nuovo test che includa le tre tipologie di domande.

a. Risultato atteso:

La creazione del test deve avvenire con successo, ciò sarà verificato sulla base di dati.

b.Metodologia:

Verrà selezionato un corso esistente e si effettuerà la creazione di un nuovo test . In seguito si eseguirà il logout e si accederà come uno degli utenti iscritti al corso.

b. Risultato atteso:

Il nuovo test deve essere visualizzato correttamente nella lista dei learning objects del corso.

2.2.6) Definizione relazioni fra learning objects:

a. Metodologia:

Verrà selezionato un learning objects di un corso ed effettuato l'inserimento di nuove precedenze.

a. Risultato atteso:

L'inserimento delle precedenze deve avvenire con successo, ciò sarà verificato sulla base di dati.

b. Metodologia:

Verrà selezionato un learning object di un corso e si tenterà di inserire il learning objects stesso nella lista delle precedenze.

b. Risultato atteso:

Il sistema deve impedire l'operazione, possibilmente a priori, non consentendo tale inserimento.

c. Metodologia:

Verrà selezionato un learning object A di un corso e si inserirà fra le sue precedenze il learning object B. In seguito si selezionerà B e si tenterà di inserire A fra le sue precedenze.

c. Risultato atteso:

Il sistema deve impedire l'operazione, possibilmente a priori, non consentendo tale inserimento.

2.2.7) Assegnamento assistente ad un corso:**a. Metodologia:**

Autenticati come Trainer, si selezionerà un corso e si selezionerà la funzione di aggiunta dell'assistente.

a. Risultato atteso:

L'assistente viene aggiunto al corso correttamente (verifica sulla base di dati.)

2.2.8) Visualizzazione risultati dei test:**a. Metodologia:**

Autenticati come Trainer, si selezionerà un corso e si selezionerà la funzione di visualizzazione dei risultati, che saranno visualizzati selezionando i vari criteri di aggregazione.

a. Risultato atteso:

I risultati sono visualizzati correttamente secondo i criteri di aggregazione.

b. Metodologia:

Si confronteranno i risultati visualizzati con i risultati presenti nella base di dati.

b. Risultato atteso:

I dati devono essere coerenti.

2.2.9) Iscrizione a corso:**a. Metodologia:**

Autenticati come studente si accederà alla pagina di iscrizione ai corsi e si effettuerà l'iscrizione ad uno di essi.

a. Risultato atteso:

L'iscrizione avviene correttamente (verificato nella base di dati).

b. Metodologia:

Autenticati come studente si accederà alla pagina di iscrizione ai corsi e si effettuerà una seconda iscrizione ad un corso a cui si è già iscritti.

b. Risultato atteso:

Il sistema deve impedire l'operazione, possibilmente a priori, non consentendo tale iscrizione.

c. Metodologia:

Si effettuerà il login con l'account del professore del corso di cui lo studente ha chiesto l'iscrizione e si controllerà la pagina delle richieste di iscrizione.

c. Risultato atteso:

La richiesta dello studente deve essere presente.

d. Metodologia:

Dalla pagina delle richieste di iscrizione si effettueranno una prova di rifiuto e una di accettazione.

d. Risultato atteso:

Nel primo caso lo studente deve visualizzare il corso nella propria home page, mentre in caso di rifiuto verrà verificato nella base di dati il mancato inserimento dello studente negli iscritti al corso.

2.2.10) Scaricamento Materiale:**a. Metodologia:**

Autenticati come studente si accederà alla pagina di un corso e si selezionerà un learning object di tipo “materiale”.

a. Risultato atteso:

Il materiale viene scaricato correttamente.

2.2.11) Sostenimento Test:**a. Metodologia:**

Autenticati come studente si accederà alla pagina di un corso e si selezionerà un learning object di tipo “test” e lo si svolgerà.

a. Risultato atteso:

Il test viene presentato correttamente e a svolgimento concluso viene correttamente inserito nel sistema (verificato nella base di dati).

b. Metodologia:

Autenticati come studente si accederà alla pagina di un corso e si selezionerà nuovamente il test appena svolto.

b. Risultato atteso:

Il sistema deve impedire l'operazione, non consentendo di svolgere nuovamente il test.

2.2.12) Visualizzazione Voti Test:

a. Metodologia:

Autenticati come studente si accederà alla pagina di visualizzazione dei risultati.

a. Risultato atteso:

I risultati sono visualizzati correttamente.

b. Metodologia:

Si confronteranno i risultati visualizzati con i risultati presenti nella base di dati.

b. Risultato atteso:

I dati devono essere coerenti.

3)Risultati dei test:

Di seguito sono riportati i risultati ottenuti dall'applicazione quando sottoposta ai casi si test precedentemente definiti.

Notazione utilizzata:

SUCCESS: indica che il test e' stato superato in modo totalmente conforme alle aspettative.

WARNING: indica che il test e' stato superato, con qualche difetto che è necessario correggere prima del rilascio ufficiale dell'applicazione.

FAIL: indica che il test non e' stato superato.

3.1)Requisiti generali:

3.1.1)Architetturali:

a. Risultato ottenuto: **FAIL**

1. Mancanza del file “application.xml”, deploy non funzionante, dopo averlo creato manualmente il deploy ha funzionato.
2. L'applicazione è visualizzata correttamente
3. Sessioni multiple funzionano correttamente.

b. Risultato ottenuto: **SUCCESS**

Dall'analisi statica del codice risulta che l'applicazione e' stata scritta in linguaggio java e utilizzando il framework Java EE.

3.1.2)Cognitivi:

a. Risultato ottenuto: **WARNING**

Seguendo alla lettera la procedura descritta nel manuale e tentando il login con un utente viene sollevata l'eccezione

`eccezione$Proxy91 cannot be cast to lexxon.session.ProfessoreManagerInterfaccia .`

Il problema è stato risolto contattando il gruppo che ha prontamente fornito risposta al problema, dovuto a una duplicazione del package dei session beans sia nell' ejb-project che nel web-project.

b. Risultato ottenuto: **SUCCESS**

L'applicazione funziona correttamente con entrambi i web browser.

3.1.3)Sicurezza:

a. **Risultato ottenuto: WARNING**

Non è possibile in generale, per un utente non autenticato, utilizzare le funzionalità del sistema. Ci sono però alcune constatazioni da fare:

- Non è impedito in alcun modo l'accesso alle pagine di amministrazione.
- Il tentativo di utilizzo di funzionalità delle pagine di amministrazione, da parte di un utente non autenticato portano sempre ad un'eccezione di tipo NullPointerException il che significa che l'operazione non è realmente impedita da una policy di sicurezza ma semplicemente viene generato un errore dovuta alla non esistenza di alcune variabili di sessione necessarie.
- Il logout non è funzionante, le variabili di sessione non vengono pulite né la sessione terminata, ciò può causare un serio problema di sicurezza nel caso di utilizzo da postazioni pubbliche.

3.2)Requisiti funzionali:

3.2.1)Registrazione:

a. **Risultato ottenuto: FAIL**

La funzionalità di registrazione, contenuta nei documenti preliminari, non è stata implementata .

b. **Risultato ottenuto: FAIL**

La funzionalità di registrazione, contenuta nei documenti preliminari, non è stata implementata .

3.2.2)Login e logout:

In conseguenza alla particolare procedura di registrazione al sistema i casi di test precedentemente definiti verranno estesi come segue:

a. **Risultato ottenuto: SUCCESS**

1. login effettuato correttamente. Viene mostrata la finestra principale dell'applicazione.
2. Il login fallisce.

b. **Risultato ottenuto: SUCCESS**

Il login fallisce.

c. **Risultato ottenuto: FAIL**

Lo stato dell'applicazione si mantiene coerente ma il logout non è funzionante. Il logout non è altro che una redirect verso la pagina di accesso ma non è un logout effettivo che termina la sessione.

3.2.3) Creazione Corso:

a. **Risultato ottenuto: SUCCESS**

La creazione del corso avviene con successo.

b. **Risultato ottenuto: SUCCESS**

Il corso è regolarmente presente nella home page del docente.

c. **Risultato ottenuto: SUCCESS**

Il corso è regolarmente presente nella lista dei corsi disponibili.

3.2.4) Aggiunta Materiale ad un corso:

a. **Risultato ottenuto: WARNING**

La funzionalità presenta una problematica: per avere successo nell'upload bisogna editare il path di salvataggio dei file nella servlet che se ne occupa. Tale problematica risulta spiegata nei casi di test forniti. Questa operazione costituisce comunque un problema nell'installazione del sistema durante la quale non dovrebbe essere necessario modificare i sorgenti delle servlet.

b. **Risultato ottenuto: WARNING**

La funzionalità presenta un grave problema. Il materiale risulta effettivamente disponibile soltanto in seguito a un deploy fatto dopo l'inserimento. Tale problematica risulta spiegata nei casi di test forniti ma non è accettabile, in casi reali, dover fare un nuovo deploy dell'applicazione ad ogni inserimento di materiale per vederne l'effetto. Inoltre l'url per la consultazione del materiale da parte di un docente non è funzionante a causa di uno "slash" contenuto nell'url al posto di un "back-slash".

3.2.5) Creazione di un test per un corso:

a. **Risultato ottenuto: FAIL**

Il test viene creato ma con due importanti problemi:

- La possibilità di inserimento è di una sola domanda, ciò non rispetta i requisiti presenti nei documenti preliminari.
- Non viene visualizzato alcun messaggio di conferma e i test non appaiono in modo chiaro nella home page, costituendo un problema di usabilità.

b. **Risultato ottenuto: FAIL**

Il test è presente ma con varie problematiche:

- Viene visualizzato un solo test, e questo test è errato perchè appartenente ad un altro corso.
- Non c'è alcun modo per accedere agli altri test creati dal docente per lo stesso corso
- Lo svolgimento dei test non è chiaro. Ciò presenta un serio problema di usabilità

3.2.6) Definizione relazioni fra learning objects:

a. **Risultato ottenuto: FAIL**

La funzionalità di definizione delle relazioni non può essere testata perchè non è stata implementata, ciò costituisce un mancato rispetto dei requisiti.

b. **Risultato ottenuto: FAIL**

La funzionalità di definizione delle relazioni non può essere testata perchè non è stata implementata, ciò costituisce un mancato rispetto dei requisiti.

c. **Risultato ottenuto: FAIL**

La funzionalità di definizione delle relazioni non può essere testata perchè non è stata implementata, ciò costituisce un mancato rispetto dei requisiti.

3.2.7) Assegnamento assistente ad un corso:

a. **Risultato ottenuto: FAIL**

L'assistente non viene aggiunto nella base di dati. Inoltre l'inserimento di un nuovo assistente utilizzando l'id dello stesso da inserire manualmente in una form risulta essere un problema di usabilità del sistema. La funzionalità sembra non essere implementata, qualunque id si inserisca nella form si ottiene una redirect verso la propria homepage.

3.2.8) Visualizzazione risultati dei test:

a. **Risultato ottenuto: FAIL**

La funzionalità di visualizzazione aggregata dei risultati non può essere testata perchè non è stata implementata, ciò costituisce un mancato rispetto dei requisiti.

3.2.9) Iscrizione a corso:

a. **Risultato ottenuto: SUCCESS**

L'iscrizione avviene correttamente, l'utente è presente nella base di dati.

b. **Risultato ottenuto: SUCCESS**

Il sistema impedisce a priori l'iscrizione a un corso a cui si è già iscritti.

c. **Risultato ottenuto: SUCCESS**

La richiesta dello studente è visualizzata correttamente dal professore.

d. **Risultato ottenuto: SUCCESS**

Il sistema si comporta in modo adeguato sia in caso di approvazione dell'iscrizione che di rifiuto della stessa da parte del professore.

3.2.10) Scaricamento Materiale:

a. **Risultato ottenuto: WARNING**

Il materiale può essere scaricato correttamente ma solamente se, dopo l'inserimento da parte del docente, è stato effettuato un nuovo deploy. Tale problematica risulta spiegata nei casi di test forniti ma non è accettabile, in casi reali, dover fare un nuovo deploy dell'applicazione ad ogni inserimento di materiale per vederne l'effetto.

3.2.11) Sostenimento Test:

a. **Risultato ottenuto: FAIL**

A causa delle problematiche riscontrate nella creazione del test non è stato possibile testare in modo approfondito questa funzionalità che sembra comunque implementata in modo errato ed incompleto.

b. **Risultato ottenuto: FAIL**

Il controllo sui test già sostenuti funziona in modo errato. Esso consente infatti, per ogni corso, l'accesso a un solo test del corso, e ad un solo studente. Impedisce così la possibilità di svolgimento di quasi tutti i test da parte di tutti gli studenti. Inoltre non è visualizzato a schermo alcun messaggio di errore specifico, il che costituisce un problema di usabilità.

3.2.12) Visualizzazione Voti Test:

a. **Risultato ottenuto: FAIL**

La funzionalità di visualizzazione aggregata dei risultati non può essere testata perchè non è stata implementata, ciò costituisce un mancato rispetto dei requisiti.

4)Conclusioni e valutazioni finali:

Il sistema non si può ritenere conforme alle specifiche nè sufficientemente aderente ai documenti preliminari consegnati. Buona parte delle specifiche risultano inoltre parzialmente o completamente non sviluppate.

La configurazione del sistema è semplice ma non aderente al manuale d'uso fornito. Il sistema risulta spesso privo, in caso di errori, di adeguati messaggi di errore, risultando così molto poco user friendly nel caso di utenti inesperti.

Alla luce di tali considerazioni, l'applicazione risulta evidentemente acerba e incompleta, non si può dunque ritenere pronta all'utilizzo in un contesto reale.