Politecnico di Milano
Facoltà di Ingegneria dell'informazione
Dipartimento di elettronica e Informazione

Documento di Testing del progetto: lesson-dn-benefico-castelnovo-castelnuovo

Cipolleschi Riccardo, Matr. 764829

Consolandi Cristian, Matr. 766167

Galli Daniele, Matr. 764838

Anno Accademico 2010-2011

Sommario

Analisi delle Funzionalità

Prima di iniziare con l'analisi delle funzionalità implementate nel progetto LESSON che si vuole testare, occorre effettuare alcune cosiderazioni preliminari. Nel documento allegato all'implementazione è descritto il funzionamento delle funzionalità principali della piattaforma. In esso è specificata la scelta di design di non occuparsi della parte di registrazione dei nuovi utenti nel sistema: questo comporta una gestione manuale, da parte del database aministrator, dell'inserimento dei nuovi utilizzatori.

Oltre a comportare un carico di lavoro, che potrebbeessere gestito in modo automatico, per l'amministratore del database, questa scelta rende il testing più complesso e meno completo, in quanto: occorre conoscere la struttura del database per poterlo modificare (inclusi i tipi associati alle colonne delle tabelle e i valori costanti che vi si possono inserire, per esempio, per discriminaretra trainer e trainee). Inoltre, sempre in relazione a questa scelta di design, l'identificatore per i nuovi utenti non è autogenerato ed è un intero. Questo implica che l'identificatore e la password devono essere comunicati dall'amministratore all'utente che l'utente debba memorizzarli. Dal momento che la gestione della registrazione non è stata gestita, non è possibile modificare autonomamente questi valori, ma ogni utente deve fare richiesta all'amministratore del sistema.

Trainer

Nel documento allegato all'implementazione, al capitolo 3, vi è una sezione dedicata al testing e ai bug non risolti. Tra i punti esposti in questa parte del documento vi sono specificate delle funzionalità del Trainer che non sono state implementate:

- La gestione dell'aggregazione dei risultati secondo diversi criteri è stata tralasciata
- Le precedenze tra learning object sono state implementate solo in moo parziale:è possibile stabilire solo relazioni di precedenza e non di opzionalità.

Queste due unzionalità non saranno quindi prese in considerazione per la parte di analisi e di test planning: si verificherà solamente che le relazioni di precedenza tra i learning object siano rispettate nel momento in cui verranno stabilite.

Dal documento di analisi dei requisiti emergono le seguenti funzionalità a disposizione di un Trainer:

- Login: un trainer, registrato nel sistema da un database administrator, deve poter accedere alla sua pagina personale.
- Crea Coso: un trainer deve poter creare un corso.
- Accettazione iscrizione: quando un trainee fa richiesta di iscriversi ad un corso, il trainer deve poterlo accettare
- Inserimento Course Material: il trainer deve poter inserire dei nuovi materiali nei corsi che gestisce.
- Specifica delle relazioni tra i Learning Object: dopo aver creato dei materiali, il trainer deve poter esprimere delle relazioni di precedenza e opzionalità tra learning object (l'opzionalità non è supportata).
- Creazione dei Checkers per i test: per ogni test che viene caricato, il trainer può associarvi ei checkers automatici oppure scegliere di correggerli manualmente.
- Correzione dei Test: il Trainer deve poter scaricare i test che deve correggere manualmente e deve poter poi caricare i risultati.
- Aggregazione dei risultati: (Non è supportata).

Per ognuna delle funzionalità che si testeranno si vuole verificare che il flusso di eventi espresso nel documento RASD sia coerente con quanto avviene nel sistema sviluppato, che il processo di inserimento dei dati sia semplice e intuitivo e che gli input siano verificati.

Si vuole inoltre verificare che le modifiche sul DB avvengano in modo coerente a dati inseriti e che si sollevino le eccezioni nei casi previsti e come sono affrontate le eccezioni.

Per poter strutturare l'analisi in modo corretto, si analizzano le funzionalità implementate a partire dai diagrammi UML riportati nel documento RASD.

Login

Per questa funzionalità non è presente nel RASD nessun Sequence Diagram né è presente lo Use Case relativo. Il Trainer comunque dovrebbe conoscere il proprio numero identificativo e la propria password e deve già essere registrato nel sistema.

Se il login ha successo, il Trainer deve condotto alla sua pagina personale.

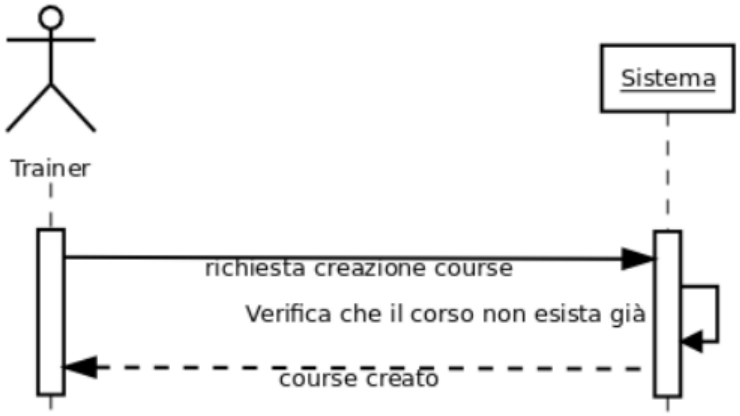
Se il login non ha successo, dovrebbe essre notificato un qualche tipo di errore che indica i motivi dell'insuccesso.

Creazione Corso

Nome dello scenario	Creazione di un course
Attori partecipanti	Trainer
Flusso degli eventi	• Il trainer inserisce i dati relativi al corso • Il trainer invia la richiesta al sistema • Il trainer riceve la conferma dell'avvenuta creazione del corso
Eccezioni	Il corso è già esistente

La descrizione testuale dell'use case è piuttosto semplice. Non ha precondizioni se non quella implicita che il trainer sia già conneso al sistema. Prevede l'inserimento dei dati relativi al corso e l'invio della richiesta di creazione del corso. E' previsto un messaggio a conferma che l'operazione si è svolta in modo corretto. L'unica eccezione prevista è il caso in cui un corso sia già esistente.

Segue il Sequence Diagram che modella l'interazione tra il Trainer e il sistema.



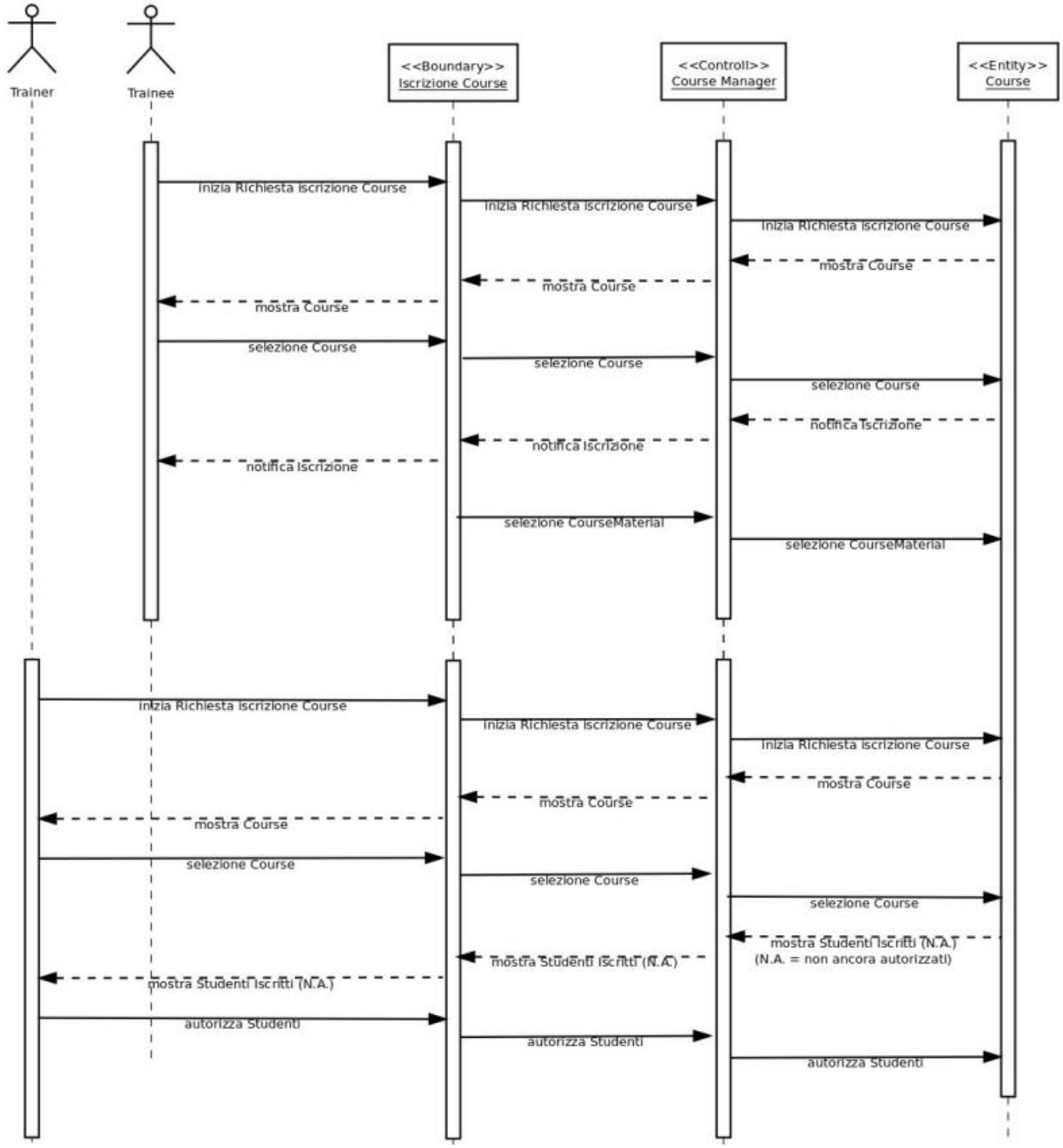
Accettazione Iscrizione:

Per questa funzionalità non è presente un flusso di eventi voltoa descriverne il funzionamento. È verosimile ipotizzare che le premesse siano:

- la presenza di un corso gestito dal trainer.
- l'esistenza di un trainee nel sistema che ha fatto richiesta di iscrizione al corso.

Al trainer dovrà essere notificato la richiesta di iscrizione e dovrà essere consentita la possibilità di accettare o rifiutare il futuro iscritto.

Questa particolare interazione con il sistema è stata raffinata nel DD del progetto con il sequence diagram seguente:



Dal sequence è possibile verificare che questa funzionalità richiede una precedente attività di un Trainee che effettua la richiesta di iscrizione ad un corso esistente. Il Trainer verifica poi l richieste in sospeso ai corsi che gestisce, ottenendo dal sistema un elenco dei corsi che hanno dei trainee in attesa e, infine, può consentire l'accesso ad un trainee.

Per questa funzionalità non sembra che siano previste eccezioni.

Inserimento Course Material:

Nome dello scenario	Caricamento course material
Attori partecipanti	Trainer

Pre-condizioni	• Esistono già:
	  ◦ il corso
	  ◦ il topic
	  ◦ la lecture
	• Il trainer deve essere titolare del corso

Flusso degli eventi	• Il trainer seleziona il course • Il trainer seleziona il topic del course material • Il trainer seleziona la lecture in cui saranno presentate • Il trainer esegue l'upload del file con il course material • Il trainer riceve la conferma dell'avvenuto inserimento del course material
Eccezioni	L'upload del file fallisce

Per questa funzionalità vi sono alcune premesse che non è chiaro di come possano essere sddisfatte: la presenza di un corso è soddisfacbile grazie alla funzionalità di creazione del corso, tuttavia come si possa effettuare la creazione di topics e di lectures non è specificato.

L'unico caso analizzato in cui il processo può fallire è il caso in cui l'upload del file non ha successo.

Creazione del checker

Per questa funzionalità non è specificato nessun Sequnce Diagram e nessun Activity Diagram. Nel seguitodel documento si fa riferimento alla possibilità di assegnare un checker ad un test ma non è descritto come può essere creato, se un checker automatico, né che caratteristiche deve avere un checker umano per poter essere assegnato ad un test.

Creazione di un test:

Nome dello scenario	Creazione test
Attori partecipanti	Trainer

Pre-condizioni	• Esistono già:
	  ◦ il corso
	  ◦ il checker (nel sistema o nel mondo esterno)
	• Il trainer deve essere titolare del corso

Flusso degli eventi	• Il trainer seleziona il course • Il trainer indica al sistema di voler creare un test • Il trainer seleziona un template • Il trainer crea il test • Il trainer fornisce un checker • Il trainer riceve la conferma dell'avvenuta creazione del test
Eccezioni	Il checker è errato per il tipo di template selezionato

Questo Use Case mostra come è possibile creare un nuovo test per un certo corso. Non è chiaro come sia possibile che un checker debba essere già presente nel sistema o nel mondo esterno: se il test da correggere deve ancora essere creato non può già esistere un checker che effettui la sua correzione, in modo automatico, nel sistema. Inoltre, nel caso il checker sia umano e quindi appartenga al mondo esterno, non sono specificati i requisiti che deve avere (in particolare potrebbe essere anche un Trainee).

Dal flusso degli eventi si deduce che il Templateè stato attribuito al test e non al checker e che l'unica eccezione sollevabile è il caso in cui il checker specificato non corrisponda al template scelto per il test.

Specifica delle Relazioni

Nome dello scenario	Creazione precedenza tra learning object
Attori partecipanti	Trainer

Pre-condizioni	• Esistono già:
	  ◦ il corso
	  ◦ almeno 2 learning object
	• Il trainer deve essere titolare del corso

Flusso degli eventi	• Il trainer seleziona il course • Il trainer indica al sistema di voler stabilire una relazione tra 2 learning object • Il trainer seleziona la relazione di precedenza (A dà la precedenza a B) • Il trainer seleziona il learning object A che deve dare precedenza • Il trainer seleziona il learning object B con precedenza • Il trainer riceve la conferma dell'avvenuta creazione della relazione

Per questo caso d'uso le premesse sono lecite e coerenti: deve già esistere un corso in cui sono presenti almeno due learning object tra cui stabilire le precedenze. Dal flusso degli eventi si deduce che il trainer

può entrare in una pagina di "creazione delle relazioni", scegliere due learning object e stailire la precedenza tra i due. Non è previsto il sollevamento di una eccezione in nessun caso: sembra quindi possibile attribuire una relazione di precedenza circolare tra learning object che impedirebbe ad qualsiasi trainee di scaricarli (se ci fosse una precedenza tra A e B e una tra B e A, un trainee T non potrebbe scaricare B perchè non possiede A ma non potrebbe scaricare A perchè non possiede B).

Inoltre questa scelta i design comporta anche la possibilità che le relazioni vengano assegnate in un momento diverso dall'upload dei materiali. Questa scelta comporta che se un trainee accedesse alla pagina di scaricamento dei materiali prima che la relazione di precedenza fosse stabilita, allora potrebbe oltrepassare la precedenza in un modo non desiderato.

Trainee

Dal documento di analisi dei requisiti emergono le seguenti funzionalità a disposizione di un Trainee:

- Login: un trainee, registrato nel sistema da un database administrator, deve poter accedere alla sua pagina personale.
- Iscrizione ad un corso: un trainee deve poter effettuare una richiesta d'iscrizione ad un corso, che potrà essere rifiutata od accettata dal trainer titolare di tale corso, ed in caso di accettazione permetterà al trainee di accedere al materiale del corso.
- Download Learning Object: un trainee deve poter scaricare il materiale presente in un corso al quale egli risulta iscritto, rispettando l'ordinamento di precedenza richiesto.
- Svolgimento test: un trainee deve poter svolgere i test presenti in un corso al quale egli risulta iscritto, rispettando l'ordinamento di precedenza richiesto.

Volendo realizzare nella fase di testing gli stessi obiettivi illustrati nella sezione sulle funzionalità del Trainer, riguardo alla coerenza con i casi d'uso specificati nella documentazione, si riporta di seguito un'analisi delle funzionalità basata sulle informazioni presenti nel documento di analisi dei requisiti.

Login

Per questa funzionalità non è presente nel RASD nessun Sequence Diagram né è presente lo scenario relativo. Il Trainee deve essere in possesso del proprio identificativo e della propria password, ricevuti in qualche altro modo al di fuori dal sistema dal database administrator, con i quali egli risulta registrato nel sistema.

Se il login ha successo, il Trainee deve condotto alla sua pagina personale.

Se il login non ha successo, dovrebbe essere notificato un qualche tipo di errore.

Iscrizione ad un corso

Per un possibile flusso di eventi relativo a questa funzionalità ci si rifà al sequence diagram raffinato nel Design Document, già mostrato nella sezione relativa al Trainer riguardante l'accettazione delle richieste d'iscrizione.

Precondizione per lo svolgimento di questa funzione è che sia presente nel sistema almeno un corso.

Il Trainee dalla propria pagina iniziale inizia la procedura di richiesta e seleziona il corso al quale desidera iscriversi. Una volta accettato potrà ricevere una notifica di questo fatto e potrà iniziare a visionare il materiale relativo al corso.

Per questa funzionalità non sembra che siano previste eccezioni.

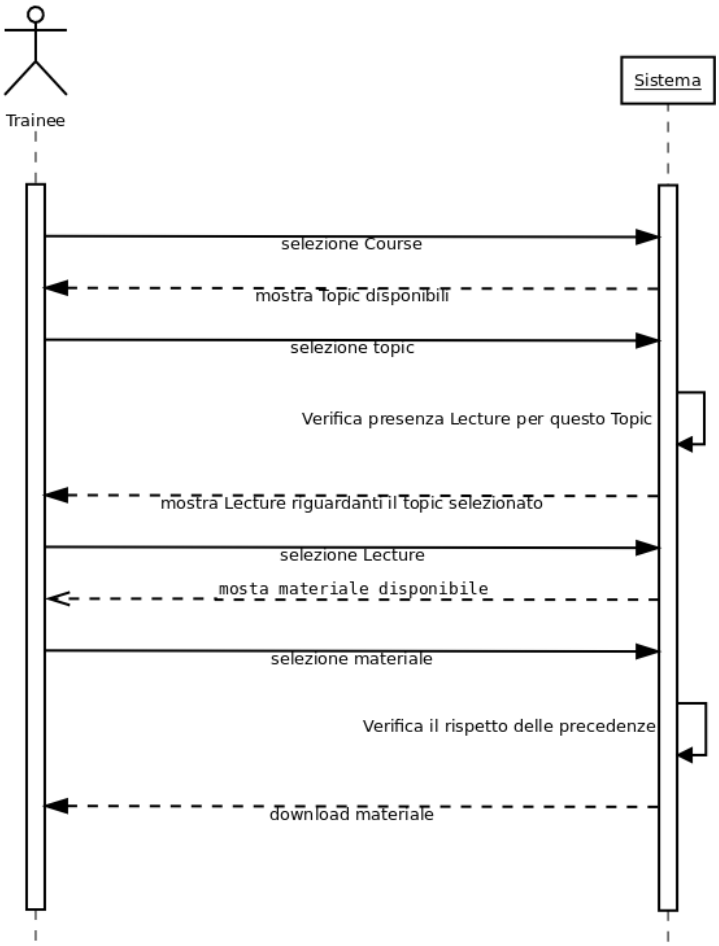
Download Learning Object

Nome dello scenario	Download course material
Attori partecipanti	Trainee
Pre-condizioni	Il trainee è iscritto al course
Flusso degli eventi	• Il trainee seleziona il course • Il trainee seleziona il topic • Il trainee seleziona la lecture • Il trainee seleziona il materiale di suo interesse • Il trainee esegue il download del materiale
Eccezioni	Il download del file fallisce

Nella documentazione il caso del download di un Learning Object è affrontato mostrando esempi di download di file relativi a Course Material, mentre non viene approfondito lo use case relativo al download di test da svolgere.

La descrizione del caso d'uso lascia sottointesa la presenza del materiale da scaricare nel sistema, organizzato secondo topic e la lecture già specificate dal trainer titolare del corso, che deve aver accettato la richiesta di iscrizione del trainee.

Di seguito il sequence diagram, che aggiunge eventi relativi al sistema riguardo al controllo della presenza di materiale e di controllo delle precedenze sui learning object.

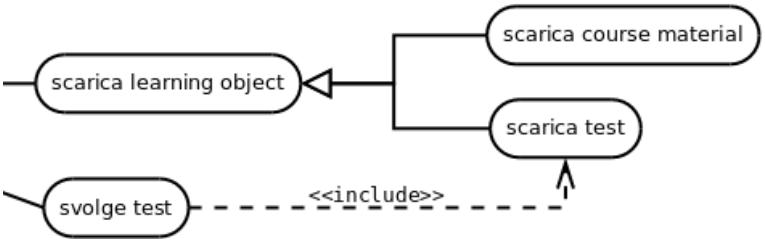


È presente anche un actvity diagram che indica la ripetizione degli eventi di selezione di topic fino a trovarne uno contente qualche lecture, e la ripetizione degli eventi di selezione del materiale fino a trovarne uno che rispetti le precedenze.

Inoltre il sequence diagram relativo a questo caso d'uso è stato raffinato anche in fase di design, mostrando nel relativo documento come i vari componenti specificati interagiscono durante il flusso degli eventi appena presentato.

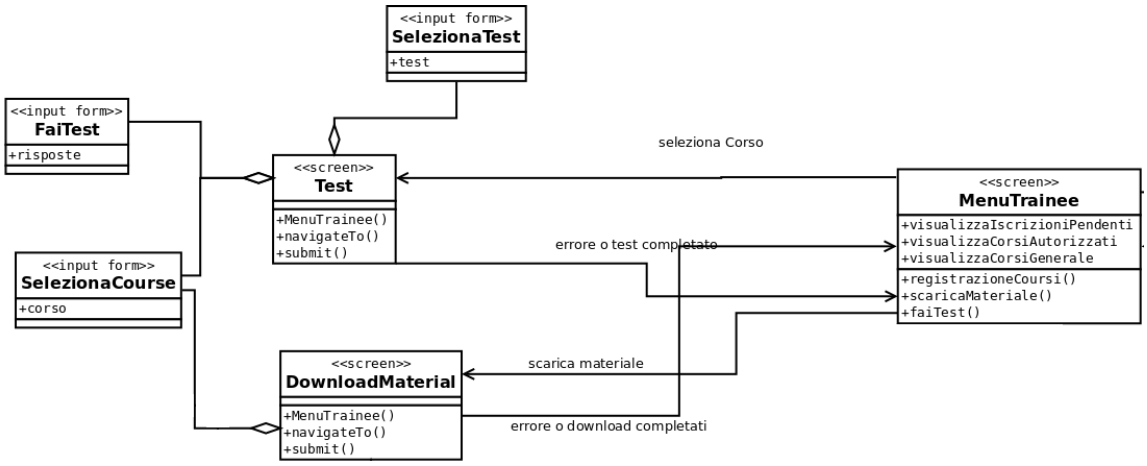
Svolgimento test

Questa funzionalità non viene approfondita nel documento RASD, nello use case diagram si indica uno use case relativo ad essa che include lo use case di scaricamento del test, specializzazione della funzionalità di download di un learning object.



Si presuppone perciò che inizialmente per svolgere un test si debba scaricare del materiale relativo ad esso.

Dal design document, dallo UX-Diagram relativo al Trainee si può poi però individuare che lo svolgimento del test vero e proprio avviene invece tramite una input form per l'inserimento di risposte alle domande del test, e la stessa cosa si evince anche dal Manuale Utente.



Non si riscontrano eccezioni se non quella relativa ad un errore generico segnalato nella schermata del test.

Assistant

Dal documento di analisi dei requisiti emergono le seguenti funzionalità a disposizione di un Assistant:

- Login
- Correzione di test

Login

Per questa funzionalità non è presente nel RASD nessun Sequence Diagram né è presente lo scenario relativo. L'Assistant deve essere in possesso del proprio identificativo e della propria password, ricevuti in qualche altro modo al di fuori dal sistema dal database administrator, con i quali egli risulta registrato nel sistema.

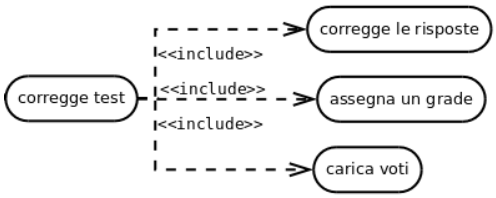
Se il login ha successo L'Assistant deve condotto alla sua pagina personale.

Se il login non ha successo, dovrebbe essere notificato un qualche tipo di errore.

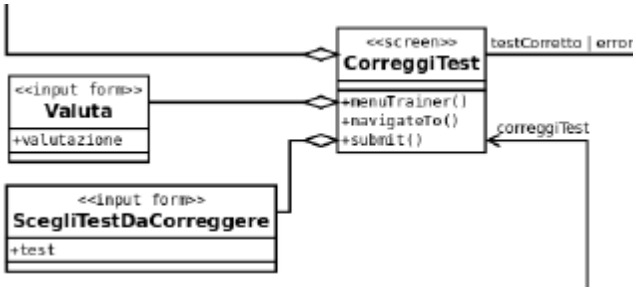
Correzione di test

Questa funzionalità dell'Assistant è disponibile anche per il trainer ed il software checker, e l'analisi svolta in questo paragrafo sarà presa in considerazione anche per il testing relativo a questi altri due attori del sistema.

Per questa funzionalità non è presente nel RASD nessun Sequence Diagram né è presente lo scenario relativo. Viene presentata attraverso lo Use Case Diagram, mostrandone tre sotto casi inclusi in esso, ovvero la correzione delle risposte, l'assegnamento di un grade ed il caricamento dei voti.



Dal Design Document, osservando lo UX-Diagram relativo al Trainer, presumendo che non vi siano differenze rispetto a come la funzionalità verrà presentata all'Assistant, si evince che la correzione avviene scegliendo dalla home dell'utente la funzionalità di correzione test, scegliendo il test tra tutti quelli presenti nel corso ed inserendo una valutazione. Non viene ben specificato come viene visualizzato od ottenuto lo svolgimento e come esso viene relazionato ad un trainee, che presumibilmente deve aver svolto il test selezionato.



Nella documentazione non viene specificato come invece si svolge in temini di flusso di eventi la correzione da parte di un software checker, e non si specificano le parti del sistema che se ne occuperanno.

Test Plan

Per effettuare il test in modo esaustivo è necessario procedure in due fasi. Nella prima fase si useranno i dati che si possono precaricare nel database tramite il link presente nella schermata di login del sistema. Questo link avvia una procedura che inserisce nel database alcuni valori di prova.

Nella seconda fase del test, i procede a una prima pulizia del database e all'inserimento manuale dei nuovi uteti con cui effettuare i test. In questo modo è possibile verificare il corretto svolgimento delle funzionalità del sistema anche a partire da una situazione pulitadel database.

Gli obiettivi, lo svolgimento, le aspettative e i componenti sollecitati dai test non cambiano rispetto a quanto esposto ne seguito e non verranno quindi ripetuti due volte.

Per verificare il corretto inserimento dei campi nel database e per poter inserire gli utenti manualmente, si utilizzeranno dei tool grafici, come MySQL Workbench e altri programmi simili, che permettono di interagire con il database.

Trainer

La pianificazione del test per l'utente di tipo Trainee si divide in due parti: la prima consste nello svolgimento dei test relativi al solo Trainer e che non comportano la partecipazione di altri attori del sistema. Una seconda parte del test prevede l'interazione con altri attori, sostanzialmente i Traiee ce sono chiamati ad interagire con il sistema pe consentire il testing delle funzionalità del trainer. Lo scopo di questi test non è quello di valutare il corretto svolgimento delle funzionalità del trainee (per le quali vi è la sezione seguente di questo stesso documento) bensì sfruttarle per verificare le funzionalità del Trainer.

Login

Obiettivo: verificare il funzionameto della funzionalità di login

Svolgimento: Per effettuare il test in modo esaustivo si efettueranno diversi tentativi.

- Uno con un id non presente nel DB (verifica degli input)
- Uno con un id che non ha tipo intero (veriica degli input)
- Uno con id intero corretto ma pwd errata (verifica degli input)
- Uno con id intero corretto e password corretta.(verifica di unzionamento corretto)

Aspettative

- Nel primo caso ci si aspetta un messaggio di errore che indichi utente sconosciuto.
- Nel secondo caso ci si aspetta un messaggio di errore differente sull'input errato.
- Messaggio simile a quello che si dovrebbe ottenere nel primo caso.
- Nell'ultimo caso l'aspettativa è quella di riuscire ad entrare all'interno del sistema.

Componenti sollecitati

- L'input form di login
- Lo screen home
- Lo screen MenuTrainer

Crea Corso

Obiettivo: verificare il corretto funzionamento dell'inserimento dei corsi.

Svolgimento

- Tentativo di inserire un corso già esistente (verifica eccezione)
- Tentativo di inserire un corso con alcuni campi vuoti (verifica degli input)
- Tentativo di inserire un corso corretto (Verifica dell'inserimento nel database)

Aspettative

- Messaggio di errore che comunica ce un corso simile è già esistente
- Messaggio di errore che impone il completamento dei campi vuoti.
- Operazione effettuata con successo.

Componenti sollecitati

- StatefulSessionBean MainTrainer con il metodo visualizzaCorsi().
- StatelessSessionBean CreaCorso con i metodi aggiungiNome() e aggiungiDescrizione().
- Lo screen MenuTrainer con il metodo creaCorso()
- Lo screen CreaCourse
- L'input form FormCourse

Inserimento CourseMaterial

Obiettivo: verificare il corretto inserimento di nuovi course material all'interno el sistema.

Svolgimento

- Tentativo di inserire un Course Material già esistente (verifica eccezione)
- Tentativo di inserire un Course Material con alcuni campi vuoti (verifica degli input)
- Tentativo di inserire un Course Material corretto (Verifica dell'inserimento nel database)

NOTA: dal RASD sembra che sia necessario anche creare un topic e una lecture prima di procedere con l'inserimento di un course material. Il test dovrà quindi verificare anche la creazione di questi elementi, seguendo i criteri enunciati sopra (inserimento di un elemento preesistente, inserimento di un elemento con campi vuoti, inserimento di un elemento corretto) e trarra risultati sul comportamento del sistema ance in questi casi. La funzionalità prevede il sollevamento di un eccezione nel momento in cui l'upload dovesse fallire: dal momento che il fallimento dell'upload non è simulabile con un test, non saràverificabile il sollevamento dell'eccezione.

Aspettative

- Nei casi in cui si tenta di inserire un elemento che già esiste ci si aspetta un messaggio di errore che invalidi l'inserimento.
- Nei casi in cui si prova ad effettuare l'inserimento di un elemento con valori nulli, ci si aspetta un messaggio di errore che porti a inserire dei valori.
- Nei casi di inserimento corretto ci si aspetta un messaggio di successo e la modifica del database.

Componenti sollecitati:

- StatefulSessionBean MainTrainer.
- StatelessSessionBean CreaLecture con i metodi selezionaCourseMaterial() e selezionaData()
- StatelessSessionBean CreaCourseMaterial con i metodi addTopic() e e selectCorso()
- StatelessSessionBean CaricaMateriale con il metodo aggiungiFile() e selezionaLecture()
- Lo Screen MenuTrainer con i metodi creaCourseMaterial(), creaLecture() e aggiungiFile().
- Lo screen creaCourseMaterial
- l'input form FormCourseMaterial
- l'input form ScegliCourse
- lo screen creaLecture
- l'input form FormLecture
- l'input form ScegliCourseMaterial
- lo screen aggiungiFile
- l'input form ScegliLecture
- l'input form CaricaFile

Creazione checkers

Poiché sembea che non vi siano delle modalità specifiche perlacreazioe dei checkers a sé stanti, si rimanda questo test al seguente che riguarda la creazione dei test: i due oggetti, infatti, sono connessi e la creazione di uno porta alla creazione dell'altro.

Creazone Test

Obiettivo: verificare il corretto inserimento dei test nel sistema.

Svolgmento: occorre verificare l'inserimento di un test per ogni template e per ogni combinazione tipo_test-tipo_checker. Per ognuna di esse si eseguiranno

- Inserimento di un test già esistente
- Inserimento di un test con campi vuoti
- Inserimento di un test con campi pieni e di un checker con camp vuoti
- Inserimento di un test corretto.

Aspettative

- Per ogni combinazione tipo_test-tipo_checker errata ci si aspetta un messaggio di errore che eviti l'inserimento.
- Per ogni tentativo i inserire un test già esistente, ci si aspetta un messaggio i errore.
- Per ogni tentativo di inserire un test cn dei valori nulli ci si aspetta un messaggio che porti a completare l'inserimento dei dati.
- L'inserimento d dati corretti dovrebbe portare ad un messaggio di inserimento avvenuto con successo e ad una modifica nel DB.

Componenti Sollecitati

- StatefulSessionBean MainTrainer
- StatelessSessionBean CreaTest con i metodi selezionaCourseMaterial(), selezionaCorso(), selezionaTemplate(), selezionaChecker()
- StatelessSessionBean Correttore con i metodi setChecker(), setCourseMaterial(), selezionaTest()
- lo screen MenuTrainer
- lo screen CreaTest
- l'input form FormTest
- l'input form ScegliCorso
- l'input form ScegliCourseMaterial

Approvazione dell'iscrizione di un trainee

Obiettivo: verificare che un trainee possa iscriversi correttamente ad uno dei corsi gestiti dal trainer.

Svolgimento: questo particolare svolgimeto si svilupperà per passi.

1. Un trainee effettua richiesta di iscrizione
2. il trainer verifica le richieste di iscrizione
3. il trainer accetta il trainee.

Si ripete qindi il test con una azione di rifiuto.

1. un nuovo trainee effettua una richiesta di iscrizione
2. il trainer verifica le richieste di iscrizione
3. il trainer rifiuta il trainee

Aspettative:

- nel primo caso il database si deve modificare rimuovendo la richiesta di iscrizione e inserendo il trainee tra gli iscritti al corso.

- nel secondo caso il database si deve modificare semplicemente rimuovendo la richiesta di iscrizione che è stata ormai servita.

Componenti sollecitati

- StatefulSessionBean MainTrainer
- StatelessSessionBean Autorizzatore con i metodi selezionaTrainee() e selezionaCorso()
- Lo screen MenuTrainer
- Lo screen autorizzaTrainee
- L'input form ScegliTraineeInAttesa
- L'input form ScegliCorsi

Specifica delle relazioni tra i Learning Objects

Obiettivo: verificare la possibilità di stabilire relazioni di precedenza tra learningObject e constatare l'impossibilità di inserire precedenze circolari.

Svolgimento:

- Tentativo di stabilire una precedenza semplice
- Tentativo di stabilire una precedenza circolare
- Tentativo di stabilire una catena di precedenze.
- Scaricamento da parte di un trainee di un learning object e, in seguito, imposizione di na precedenza sul learning object.

Aspettative

- Il primo test dovrebbe essere possibile e non dovrebbe portare a situazioni di errore
- Nel second test, il sistema non dovrebbe consentire al Trainer di concludere l'operazione poichè una precedenza circolare impedirebbe lo scaricamento del file
- Il terzo test dovrebbe riuscire con successo.
- Per come sembra implementato il sistema, il quarto test non dovrebbe presentare problemi.

Componenti Sollecitate

- StatefulSessionBean MainTrainer
- StatelessSessionBean CreaTest con il metodo aggiungiPrecedenza()
- StatelessSessionBean CreaCourseMaterial con il metodo addPrecedenza()
- Lo screen MenuTrainer.

Correzione Test

Obiettivo: verificare che vi sia la possibilità di correggere un test e i asegnare un voto al trainee.

Svolgimento: per questo test si effettueranno 3 correzioni differenti

1. Correzione manuale: scaricamento di un test che prevede la correzione del Trainer in prima persona
2. Correzione automatica(risposta Multipla): la correzione di un test a risposta multipla
3. Correzione automatica(JUnit): la correzione di un test che ha associato un checker JUnit.

Aspettative

- Le tre correzioni dovrebbero portare delle modifiche al database per quanto riguarda i voti dei trainee che hanno sostenuto il test.

Componenti Sollecitati

- StatefulSessionBean MainTrainer
- StatelessSessionBean Correttore con il metodo correggi()

- Lo screen MenuTrainer
- Lo Screen CorreggiTest
- L'input form Valuta
- L'input form ScegliTestDaCorreggere
- L'input orm Scegli Course.

NOTA: per poter eseguire questo test è necessario che almeno un trainee abbia sostenuto il test. Occorre quindi avere a disposizione almeno tre svolgimenti: uno per un test che prevede un checker umano, uno per un test che prevede un checker a risposta multipla e uno per un test che prevede un checker JUnit.

Trainee

Si deliniano qui di seguito i casi di test che verranno effettuati sul sistema per verificare l'implementazione delle varie funzionalità relative ad un utente di tipo Trainee individuate precedentemente, basandosi sull'analisi della documentazione svolta nello specifico per ognuna di esse.

Login

Obiettivo: verificare il funzionamento della funzionalità di login per il Trainee
Svolgimento: Verranno eseguiti i seguenti casi di test:

- Tentativo di accesso con Id di utente di tipo non intero (verifica degli input)
- Tentativo di accesso con Id di utente non presente nel sistema (uno o più utenti registrati nel sistema) (verifica degli input)
- Tentativo di accesso con Id di utente non presente nel sistema (uno o più utenti registrati nel sistema) (verifica degli input)
- Tentativo di accesso con Id di utente di tipo Trainee presente nel sistema ma con password errata (verifica degli input)
- Tentativo di accesso con Id di utente di tipo Trainee presente nel sistema con password corretta(verifica di funzionamento corretto)

Aspettative

- Nei primi tre casi ci si aspetta un errore, o comunque di non riuscire ad entrare nel sistema
- Nell'ultimo caso l'aspettativa è quella di riuscire ad entrare all'interno del sistema.

Componenti sollecitati

- L'input form di login
- Lo screen home
- Lo screen MenuTrainee

Iscrizione ad un corso

Obiettivo: verificare la possibilità di richiesta di un Trainee dell'iscrizione ad un corso ed il mantenimento della coerenza nelle azioni successivamente disponibili per il Trainee in base all'esito (accettazione o rifiuto)
Svolgimento: Verranno eseguiti i seguenti casi di test:

- Richiesta di iscrizione ad un corso senza che avvenga l'accettazione da parte del Trainer titolare del corso
- Richiesta di iscrizione ad un corso con accettazione da parte del Trainer titolare del corso

Aspettative

- In entrambi i casi ci si aspetta che la richiesta vada a buon fine e che si possa risalire alla richiesta in stato d'attesa prima che il Trainer accetti o meno
- Ci si aspetta poi di poter accedere o meno al corso della richiesta e di poter visualizzare lo stato corretto della richiesta, a seconda dell'accettazione o meno da parte del trainer.

Componenti sollecitati

- Lo Stateful Session Bean MainTrainee ed i suoi metodi visualizzaIscrizioniPendenti(), visualizzaCorsiAutorizzati() e visualizzaCorsiGenerale()
- Il Session Bean Iscrittore ed il suo metodo selezionaCorso()
- Lo screen MenuTrainee ed il suoo metodo registrazioneCorsi()
- Lo screen registrazioneCorsi
- L'input form SelezionaCourse

Download Learning Object

Obiettivo: verificare la possibilità di richiesta di un Trainee di effettuare il Download di un file caricato da un Trainer in un corso al quale il Trainee risulta iscritto

Svolgimento: Verranno eseguiti i seguenti casi di test:

- Tentativo di download di un file relativo ad un Course Material precedentemente caricato in qualità di trainer titolare di un corso al quale il Trainee risulta iscritto per il quale il Trainee non ha soddisfatto le precedenze
- Download di un file relativo ad un Course Material precedentemente caricato in qualità di trainer titolare di un corso al quale il Trainee risulta iscritto e per il quale il Trainee ha soddisfatto le precedenze

Aspettative

- Nel primo caso ci si aspetta la notifica del fatto che alcune precedenze non sono state rispettate
- Nell'ultimo caso ci si aspetta di poter recuperare il file, che questo sia utilizzabile e sia identico al file precedentemente caricato

Componenti sollecitati

- Il Session Bean Stateful MainTrainee
- Il Session Bean Scaricatore e tutti i suoi metodi
- Lo screen MenuTrainee ed il suo metodo scaricaMateriale()
- Lo screen DownloadMaterial
- Le input forms SelezionaCourse, SelezionaCourseMaterial, SelezionaLecture e SelezionaFile

NOTA: data l'impossibilità di simulare l'errore in fase di download non viene testata la gestione di tale eccezione

Svolgimento test

Obiettivo: verificare la possibilità di un Trainee di effettuare un Test creato da un Trainer in un corso al quale il Trainee risulta iscritto.

Svolgimento: Verranno eseguiti i seguenti casi di test:

- Tentativo di effettuare un Test multiple choice precedentemente creato in qualità del Trainer titolare di un corso al quale il Trainee risulta iscritto
- Tentativo di effettuare un Test a risposta aperta precedentemente creato in qualità del Trainer titolare di un corso al quale il Trainee risulta iscritto
- Tentativo di effettuare un Test con checker di tipo JUnit precedentemente creato in qualità del Trainer titolare di un corso al quale il Trainee risulta iscritto

Aspettative :

- In tutti i casi ci si aspetta di poter svolgere il test selezionato, che questo corrisponda a quello precedentemente creato e che nel caso di valutazione automatica, la correzione vada a buon fine e si riscontri nel database un voto corretto (non essendo specificato nella documentazione, però, non è da subito chiaro come esso sarà assegnato).

Componenti sollecitati

- Il Session Bean Stateful MainTrainee
- Il Session Bean FaiTest ed i suoi metodi selezionaCorso() e selezionaTest()
- Il Session Bean correttore ed il suo metodo correggi
- Lo screen MenuTrainee ed il suo metodo faiTest()
- Lo screen Test
- Le input forms SelezionaCourse e SelezionaTest

Assistant

Si deliniano qui di seguito i casi di test che verranno effettuati sul sistema per verificare l'implementazione delle varie funzionalità relative ad un utente di tipo Assistant individuate precedentemente, basandosi sull'analisi della documentazione svolta nello specifico per ognuna di esse.

Login

Obiettivo: verificare il funzionamento della funzionalità di login per l'Assistant
Svolgimento: Verranno eseguiti i seguenti casi di test:

- Tentativo di accesso con Id di utente di tipo non intero (verifica degli input)
- Tentativo di accesso con Id di utente non presente nel sistema (uno o più utenti registrati nel sistema) (verifica degli input)
- Tentativo di accesso con Id di utente non presente nel sistema (uno o più utenti registrati nel sistema) (verifica degli input)
- Tentativo di accesso con Id di utente di tipo Assistant presente nel sistema ma con password errata (verifica degli input)
- Tentativo di accesso con Id di utente di tipo Assistant presente nel sistema con password corretta(verifica di funzionamento corretto)

Aspettative

- Nei primi tre casi ci si aspetta un errore, o comunque di non riuscire ad entrare nel sistema
- Nell'ultimo caso l'aspettativa è quella di riuscire ad entrare all'interno del sistema.

Componenti sollecitati

- L'input form di login
- Lo screen home

Correzione Test

(I casi di test relativi a questa funzionalità sono simili a quelli svolti per la funzionalità analoga del trainer, limitatamente al caso della correzione manuale)

Obiettivo: verificare che vi sia la possibilità di correggere un test e i assegnare un voto al trainee.

Svolgimento Verranno eseguiti i seguenti casi di test:

- Correzione manuale di un Test per cui l'Assistant è stato precedentemente indicato come correttore dal Trainer creatore del Test

Aspettative

- La correzione dovrebbero portare delle modifiche al database per quanto riguarda i voti dei trainee che hanno sostenuto il test.

Componenti Sollecitati

Nel Design Document non sono specificati i componenti relativi all'Assistant,si assume che l'architettura sollecitata sia simile a quella delineata per l'analoga funzionalità del Trainer

Risultati dei Test

Nella sezione che segue verranno raccolti i risultati dei test effettuati sul sistema analizzato. Per ogni tipo di utente sono state testate le funzionalità analizzate in precedenza secondo i criteri esposti nel test plan. A seguito di ogni funzionalità vi possono essere delle considerazioni sull'implementazione testata.

Trainer

Test sul login:

Con i dati precaricati

Test con un id non presente nel DB (verifica degli input):

- Nessun messaggio di errore, ma si rimane nella pagina di login che viene pulita.
- Stesso risultato con id intero ma campo password lasciato vuoto.

Test con un id che non ha tipo intero (veriica degli input):

- Internal error. `HTTP Status 500. NumberFormatException` non catturata.
- Lasciando il campo id vuoto si ottiene lo stesso errore precedente.

Test con id intero corretto ma pwd errata (verifica degli input):

- Nessun messaggio di errore, si rimane nella pagina di login che viene pulita

Uno con id intero corretto e password corretta.(verifica di unzionamento corretto)

- Si entra correttamente nella schermata principale del Trainer

A partire da un DB pulito

Si ottengono gli stessi risultati del caso precedente.

Test sulla creazione di un corso

Con i dati precaricati

Tentativo di inserire un corso già esistente (verifica eccezione)

- Non viene effettuato nessun controllo. Il trainer può creare tanti corsi uguali e tutti con lo stesso nome.

Tentativo di inserire un corso con alcuni campi vuoti (verifica degli input)

- Nome inserito e descrizione nulla: la creazione avviene con successo
- Tutti i campi vuoti: il sistema porta ad una pagina vuota, non ci sono controlli e l'utente non può fare nulla.
- Nome vuoto e descrizione casuale: il sistema porta alla medesima pagina bianca precedente.

Tentativo di inserire un corso corretto (Verifica dell'inserimento nel database)

- Funzionamento Corretto

A partire da un DB pulito

Si ottengono gli stessi risultati del caso precedente.

Test sull'inserimento di un Course Material

Tentativo di inserire un Course Material già esistente (verifica eccezione)

- Inserimento consentito senza alcuna eccezione
- Se un trainer crea due corsi con lo stesso nome, la combobox course presenta l'elenco di corsi che il trainer gestisce con i corsi eventualmente ripetuti, ma senza dare informazioni su quali sono (per esempio se un Trainer crea due corsi IS2, la combo box li mostra entrambi ma senza dare ulteriori informazioni per discriminare.)

Tentativo di inserire un Course Material con alcuni campi vuoti (verifica degli input)

- Nessun controllo sul nome del topic, è possibile inserire dei course material senza topic.

Tentativo di inserire un Course Material corretto (Verifica dell'inserimento nel database)

- Tutti gli inserimenti avvengono con successo.

NOTA: Differentemente con quanto esposto nel RASD, alla creazione di Course material non si possono caricare File e non ci sono specificate le lectures a cui i Course Materials sono associati.

A partire da un DB pulito

Si ottengono gli stessi risultati del caso precedente.

- **Sottotest sulla creazione di Lectures**

Per creare le lecture occorre entrare nell'opportuno menu. Anche in questo caso, i corsi duplicati sono sempicemente elencati senza dare inormazioni in modo da potrli discriminare.

Alla creazione della lecture è necessario impostare una data in cui si terrà la lecture:

- Non c'è un controllo sui valori: il sistema dice che la lecture è creata con successo ma il DB non si aggiorna correttamente.
- E' possibile creare delle lecture con il campo course material associato vuoto solo se non ci sono course material associati a quel corso.
- Lasciando tutti i campi vuoti si genera un `NumberFormatException` dovuto ai campi della data.
- E' possibile creare lecture precedenti alla data corrente

NOTA: Anche la creazione di lecture non consente il caricamento dei file. Per poter caricare un file è necessario accedere all'apposita funzione Aggiungi Files.

A partire da un DB pulito

Sono state evidenziate le seguenti situazioni particolari:

- Se si crea una lecture con una data non valida (ad esempio il 35/10/2010) il sistema annuncia che la lecture è creata con successo in quella stessa data. Nel DB la lectures corrispondente presenta una data differente (per l'esempio: 05/12/2010).
- Se si crea una lecture con una data non valida (ad esempio il 10/15/2010) il sistema annuncia che la lecture è creata con successo in quella stessa data. Nel DB la lectures corrispondente presenta una data differente (per l'esempio: 10/04/2011)
- A parte queste diversità, il funzionamento è il medesimo che nel caso di database già inizializzato.

- **Sottotest sull'Invio dei File**

Per poter inviare dei file è quindi necessario accedere specificatamente alla funzionalità omonima. Si sono riscontrati i seguenti problemi:

- Se non si specifica quale file si vuole inviare si ottiene una `FileNotFoundException` che non è gestita dal sistema.
- Se si seleziona un corso per cui non ci sono learning object non viene emesso alcun messaggio di errore: il sistema conduce ad una pagina in cui viene mostrato l'elenco (vuoto) di course material tra cui scegliere e i comandi per tornare indietro.
- Se si seleziona un corso in cui è presente un course material senza nome (possibilità che si può ottenere) la schermata di selezione dei course material presenta una lista con un elemento senza nome. L'elemento dovrebbe essere un link ma poichè la stringa ha lunghezza 0 non è selezionabile e non si può procedere con l'upload del file.
- Se si cerca di caricare un file associato ad un course material di un corso per cui non sono state create lectures, si ottiene una `NullPointerException`.

- Il file viene salvato con successo SE: è presente il corso, è presente il Course Material, è presente la lecture, il sistema trova il file. Il file viene salvato in una cartella temporanea del server: risulta complicato tenere tracia dei caricamenti dei file da parte dei trainer.

A partire da un DB pulito

Si ottengono gli stessi risultati del caso precedente.

CONSIDERAZIONI: il caricamento di un course material è molto frammentato, occorre seguire una procedura che non è guidata e ciò potrebbe portare gli utenti a procedere in un ordine che non è quello corretto, ottenendo errori e risultati indesiderati.

Test sulla creazione di un Test

Inserimento di un test già esistente

- La creazione del test impone la scelta di un template e di uno dei due tipi di checker.
- La selezione dei course material trattati mostra l'identificativo all'interno del DB (che l'utente non può conoscere) e il topic trattato (eventualmente nullo): può capitare quindi di avere un elenco di numeri senza significato. Inoltre ad un Trainer sono visibili i course material caricati da altri Trainer.
- Se non si sceglie alcun CourseMaterial su cui fare il test, viene sollevaa l'eccezione `NullPointerException` che non è stata gestita.
- La selezione del template `MUTIPLECHOICE` e a scelta del correttore software genera l'eccezione: `javax.ejb.EJBException: java.lang.IllegalArgumentException`

A partire da un DB pulito

Sono state riscontrate le seguenti particolarità:

- In assenza di template, è possibile proseguire nella creazione di un test con i checker umano senza incorrere in alcun tipo di errore. Tuttavia la procedura non viene portata a termine e si ottiene al termine l'eccezione `NumberFormatException: For input string: ""`
- In assenza di template, il tentativo di creare un test con checker software presenta l'eccezione `NumberFormatException: null`.
- In presenza di template, se non si completano i campi si solleva un'eccezione di `NullPointerException`.
- Non è possibile creare alcun tipo di test con correttore software: si presenta sempre l'eccezione `javax.ejb.EJBException: java.lang.IllegalArgumentException: org.hibernate.QueryException: could not resolve property: tipo of: it.polimi.lesson.entity.test.SwChecker` oppure l'eccezione `java.lang.NumberFormatException: For input string: ""`

- **Sottotest sulla Creazione di un Template**

Il sistema presenta la funzionalità di creazione di un nuovo template, funzionalità non esposa in alcun documento.

E' possibile creare una nuova domanda o un template da domande esistenti.

Nel caso di una nuova domanda, questa può essere multipleChoice. Il sistema quindi propone i seguenti passi di creazione:

1. chiede se si vuole creare una nuova "choice" o se si vuole creare un multipleChoice a partire da una "multiple choice" esistente.
2. Se si sceglie crea nuova choice, compare una form che consente di inserire un'opzione. Inserendo un valore testuale viene aggiunta una scelta.

3. È quindi possibile inserire una domanda e scegliere quali sono le alternative disponibili tra tutte le alternative presentate nel sistema (anche da parte di altri Trainer.)

Sono stati riscontrati i seguenti problemi:

- Nella pagina di creazione di MultipleChoice, il titolo presentato è "Autorizza Trainee" e viene chiesto di inserire una domandae di selezionare quali saranno le opzioni disponibili.
- Se si lascia la domanda vuota, il sistema procede comunque, chiedendo quale è la risposta corretta (il titolo rimane: Autorizza Trainee).
- Selezionando la risposta corretta si accede alla pagina Domanda Creata.
- Cercando di inserire un'opzione già presente nel DB si ottiene l'eccezione: `javax.ejb.EJBException: javax.persistence.PersistenceException` non gestita al sistema.

Nel caso in cui il test scelto sia OpenQuestion, si accede ad una form in cui è possibile inserire la domanda e la risposta desiderata. I problemi evidenziati sono:

- Lasciando entrambi i campi vuoti si accede ad una pagina completamente bianca.
- Lasciando il campo domanda vuoto si accede ad una pagina completamente bianca.
- Lasciando il campo risposta desiderata vuoto si ottiene la creazione della domanda.

Nel caso di un test che presenta un esercizio, viene presentata una text area in cui è possibile inserire il testo dell'esercizio.

- Se si lascia l'area di testo bianca si accede ad una pagina totalmente bianca e senza controlli.
- Se si completa l'area di testo si accede alla schermata di esercizio creato con successo.
- Risulta consentito l'inserimento di esercizi con lo stesso testo.

Terminata la creazione delle singole domande, è possibile finalmente creare il template. Vi sono diversi tipi di template a disposizione:

1. Un template per le risposte multiple:
   - Se non si sceglie nessuna domanda da inserire nel template si ottiene un'eccezione `NullPointerException` non gestita.
   - La scelta delle domande nel test porta alla creazione del template.
2. Un Template per le domande aperte:
   - Se non si scelgono le domande da inserire nel template si ottiene un'eccezione `NullPointerException` non gestita.
   - La scelta delle domande porta alla creazione del template
3. Un template per gli esercizi:
   - Se non si scelgono gli esercizi da inserire nel template si ottiene un'eccezione `NullPointerException` non gestita.
   - La scelta degli esercizi porta alla creazione del template

Si è poi riscontrata la possibilità di creare JUnitChecker separatamente da un Test. Qui il caricamento del correttore da file Jar va a buon fine, ed è poi possibile assegnare ad un esercizio il correttore JUnit.

A partire da un DB pulito

Anche in questo caso occorre considerare due casi: la creazione delle singole domande e la creazione del template vero e proprio. Si propone prima il testing sulla creazione delle singole domande e i risultati ottenuti e poi il testing sui template, con i rispettivi risultati.

Risposte Multiple: la creazine delle opzioni funziona come nel caso precedente. Inoltre:

- Creando una nuova domanda e lasciando tutti i campi vuoti si ottiene l'eccezione `NullPointerException.`
- Si procede con la creazione della domanda anche lasciando il testo della domanda vuoto.

Risposta aperta: funziona come il caso precedente.
Esercizio: funziona come nel caso precedente.

Creazione del Template: multiplechoice

- Se non ci sono scelte o se non si effettua nessuna selezione presenta l'eccezione `NullPointerException`.

Creazione del Template: openQuestion

- Se non ci sono scelte o se non si effettua nessuna selezione presenta l'eccezione `NullPointerException`.

Creazione del Template: esercizio

- Se non ci sono scelte o se non si effettua nessuna selezione presenta l'eccezione `NullPointerException`.

CONSIDERAZIONI: la creazione di un template per un test risulta particolarmente complicata. Il processo prevede: la creazione delle alternative alle domande, la creazione delle domande e l'associazione delle alternative già create alla domanda che si vuole inserire nel test e, infine, la creazione del test scegliendo le domande che lo comporranno. Sebbene quanto descritto possa sembrare semplice, la procedura non è specificata in alcun documento. Inoltre la scelta delle alternative per una domanda avviene da un elenco di possibilità popolato da tutti i trainer che utilizzano il sistema, così come la scelta delle domande che possono comporre il test non è limitata alle sole domande create dal trainer che sta creano il test. Non viene assicurata quindi l'isolatezza degli utenti che possono interagire con i dati degli altri.

Test sulla creazione di Relazioni tra i Learning Object

Tentativo di stabilire una precedenza semplice:

- E' possibile stabilire le precedenze semplici

Tentativo di stabilire una precedenza circolare

- Come temuto, è possibile stabilire delle relazioni circolari. Questo comporta l'impossibilità da parte di un trainee di scaricare dei materiali.

Tentativo di stabilire una catena di precedenze.

- E' possibile creare una catena di pecedenze.

Scaricamento da parte di un trainee di un learning object e, in seguito, imposizione di na precedenza sul learning object.

- A causa di un bug nella parte dedicata al trainee non è possibile effettuare lo scaricamento di alcun tipo di materiale. Pertanto, questo specifico caso di test, sebbene importante, non può essere verificato (vedere la sezione di questo stesso documento dedicata ai Trainee per maggiori dettagli.)

<u>A partire da un DB pulito</u>

Si ottengono gli stessi risultati del caso precedente.

NOTA: Per come è stato realizzato il sistema, le precedenze possono essere stabilite tra learning object appartenenti a corsi diversi e anche tra Learning Object creati da trainer diversi. Ovviamente questa situazione non dovrebbe essere consentita. Non è possibile inoltre verificare il corretto funzionamento dei

downloads: nella schermata di scaricamento dei file dei Trainee non vi sono controlli per avviare il download (vedere la parte sul testing del trainee per maggiori dettagli.)

Test sull'autorizzazione dei Trainee

- Se non si sceglie il trainee da autorizzare si ottiene una `IllegalStateException` non gestita.
- Non c'è il pulsante di rifiuto
- L'autorizzazione funziona nel modo corretto se si sceglie il trainee da autorizzare.

A partire da un DB pulito

Si ottengono gli stessi risultati del caso precedente.

Test sulla correzione dei Test

- Dopo che il trainee ha scelto quale test effettuare e ha inviato la risposta, il trainer può accedere alla funzionalità di correzione dei test, scegliendo il test da correggere e inserendo la valutazione.
- Se si lascia il campo valutazione vuoto, si assegna con successo una valutazione al test e la pagina che viene mostrata ha come titolo Course Creato. Il voto che viene assegnato in questo caso è -1.
- E' possibile correggere due volte lo stesso test
- Se si assegna un altro voto ad un test già corretto questo sovrascrive il precedente
- La correzione automatica attraverso un JUnitChecker avviene correttamente su piattaforma Linux, mentre presenta un errore su piattaforma Windows (vedi test di svolgimento Test nella sezione relativa al Trainee)

A partire da un DB pulito

Nel caso in cui un trainer desideri effettuare più correzioni consecutive dei test si ottiene l'eccezione: `org.hibernate.LazyInitializationException: failed to lazily initialize a collection of role: it.polimi.lesson.entity.test.TestSvolto.risposteChiuse, no session or session was closed`

Trainee

Login

Risultati dei test:

- Inserendo una stringa o non inserendo alcun valore al posto dell'id di tipo intero viene lanciata un'eccezione non gestita
- Tentando l'accesso con id non esistenti o id esistenti e password errate non si riesce ad accedere al sistema, non vi è segnalazione dell'errore, ma solo il ricaricamento della pagina d'accesso
- Accedendo con Id utente e password corretta il funzionamento corretto viene rispettato, accedendo alla pagina dello specifico utente.

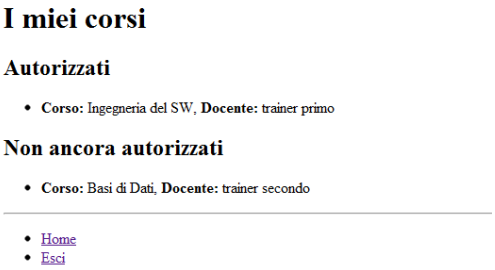
Conclusioni:

La funzionalità principale di accesso è stata implementata correttamente, manca rispetto alle aspettative una gestione delle eccezioni relative al tipo di dati in input e la segnalazione di errori dovuti alla non correttezza delle informazioni inserite.

Iscrizione ad un corso

Risultati dei test:

- Effettuando una richiesta di iscrizione ad un corso questo viene visualizzato correttamente tra i corsi non ancora autorizzati, secondo le aspettative, e non è possibile accedere ai files o ai test di tale corso
- Una volta accettata la richiesta da parte del Trainer che gestisce il corso, questo appare tra i corsi autorizzati ed è possibile visualizzare il corso tra quelli da cui è possibile scegliere per effettuare il download di files o svolgere test, secondo le aspettative.



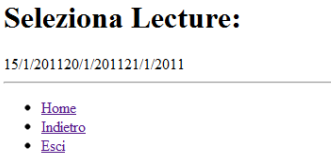
Conclusioni:

Gli aspetti relativi a questa funzionalità presentati nella documentazione sono stati implementati correttamente.

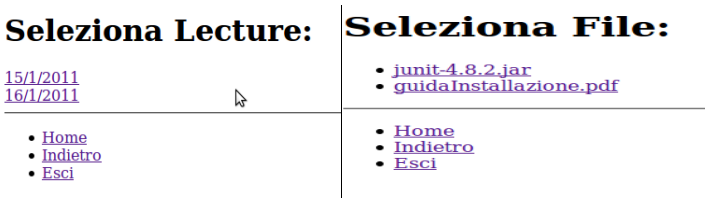
Download Learning Object

Risultati dei test:

- Per un probabile errore in fase di creazione della pagina di selezione della lecture, la scelta di questa non è possibile dato che non viene mostrato nessun link per la selezione, rendendo impossibile il download dei files ad essa abbinati.



- Il test precedente è stato realizzato su piattaforma Windows. A seguito dell'errore riscontrato si è cercato di svolgere il test anche su piattaforma Linux, in cui il sistema è stato sviluppato. In questo caso si è verificato invece un corretto funzionamento del download dei files, la scelta delle lectures avviene correttamente, tutte quelle inserite dal Trainer sono visualizzate, così come la lista dei files. I file possono essere scaricati e corrispondono a quelli inseriti dal Trainer.



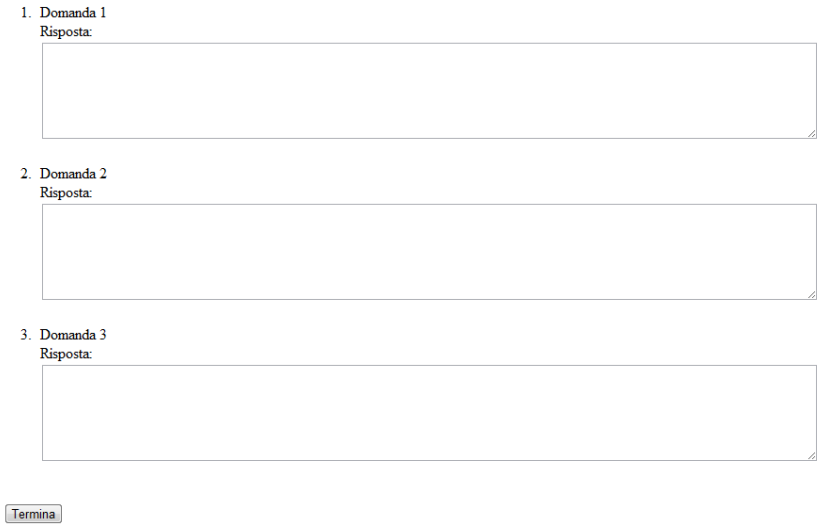
Conclusioni

Su piattaforma Windows, a causa di un bug nella schermata di scelta della lecture relativa al file da scaricare non è stato possibile testare il download dei files e la verifica delle precedenze. Su Linux la funzionalità risulta implementata correttamente.

Svolgimento test

Risultati dei test:

- Lo svolgimento del test multiple choice precaricato dalla funzionalità per il popolamento automatico del database funziona correttamente, permettendone la risposta all'unica domanda ed avviando la correzione automatica da parte del software checker ad esso abbinato. La correzione avviene secondo le aspettative ed assegnando una valutazione proporzionale. Lo svolgimento di altri test di tipo multiple choice riesce ad avvenire impostando per questi checker umani, alla fine però non si ha la valutazione ottenuta. La creazione di Test multiple choice con checker automatici non si riesce ad ottenere, e per questo non è possibile testarne lo svolgimento.
- Lo svolgimento dei test a risposta aperta avviene secondo le aspettative, mostrando tutte le domanda inserite dal Trainer e permettendone il completamento delle risposte da parte del Trainee. Essendo il correttore di questo tipo di test umano non si può conoscere la valutazione dello svolgimento. Si è riscontrato però che inserendo nelle risposte un numero eccessivo di caratteri, viene sollevata una eccezione non gestita. Inoltre, una volta inviato il test, utilizzando il comando indietro del browser si riesce a tornare alla finestra di svolgimento del test ed inviare più di uno svolgimento per lo stesso Trainee, cosa che invece non viene permessa nel normale utilizzo del sistema.



- Lo svolgimento di un test di con correttore JUnit su piattaforma Windows non va a buon fine, poiché quando si tenta di caricare il file contente le classi java da testare viene sollevata dall'applicazione la seguente eccezione: `java.lang.ClassNotFoundException: org.junit.runner.JUnitCore`.
- Lo svolgimento di un test di con correttore JUnit è stato ripetuto su piattaforma Linux. In questo caso invece, la procedura va a buon fine, mostrando la valutazione ottenuta tramite la correzione automatica. Questa è stata testata in 2 casi e mostra per il caso in cui lo svolgimento soddisfa i test JUnit richiesti la valutazione massima impostata dal Trainer, mentre la valutazione è pari a zero per uno svolgimento che presenta il fallimento di un test JUnit.

Conclusioni

Su piattaforma Windows si è riusciti a svolgere il test a risposta aperta secondo le aspettative, ed il test a risposta multipla solo nel caso del test già caricato sul sistema, mentre le altre funzionalità testate hanno portato a verificarsi degli errori che ne hanno compromesso l'utilizzo da parte dell'utente. Su piattaforma Linux invece si è riusciti anche a svolgere test con correttore JUnit. Questo indica probabilmente che l'errore è dovuto alla diversa gestione dei files e dei filepath nei due diversi sistemi operativi.

Assistant

Login

Risultati dei test:

- Inserendo una stringa o non inserendo alcun valore al posto dell'id di tipo intero viene lanciata un'eccezione non gestita
- Tentando l'accesso con id non esistenti o id esistenti e password errate non si riesce ad accedere al sistema, non vi è segnalazione dell'errore, ma solo il ricaricamento della pagina d'accesso
- Accedendo con Id utente e password corretta il funzionamento corretto viene rispettato, accedendo alla pagina dello specifico utente.

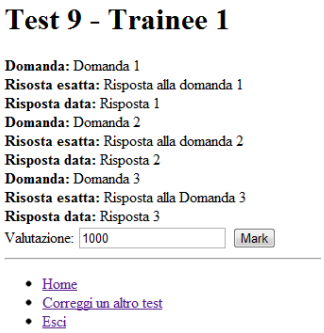
Conclusioni:

La funzionalità principale di accesso è stata implementata correttamente, manca rispetto alle aspettative una gestione delle eccezioni relative al tipo di dati in input e la segnalazione di errori dovuti alla non correttezza delle informazioni inserite.

Correzione Test

Risultati dei test:

- Inserendo tramite il Trainer creatore del Test l'Assistant per cui si sta svolgendo il testing come correttore, e inserendo nel sistema alcuni svolgimenti di Trainee, essi vengono visualizzati correttamente dall'Assistant, e viene permesso ad egli di inserirne la valutazione relativa nel sistema. Non avviene però un controllo sul valore inserito come voto, che può superare il parametro che indica il voto massimo raggiungibile nel Test in fase di creazione da parte del Trainer, e che viene comunque memorizzato nel database. Inoltre inserendo una stringa al posto di un valore numerico viene sollevata una eccezione non gestita.



- Il sistema permette di indicare tramite il Trainer creatore di un Test l'Assistant come correttore umano di un Test con Template di tipo Multiple Choice. Quando si accede con l'Assitant per correggere tale Test viene sollevata un'eccezione nel momento in cui lo si seleziona. Inoltre, se si assegna un Assistant come correttore di un Test con Template di tipo Multiple Choice, si verifica che nella lista dei Test da correggere non saranno visualizzati i Test successivamente assegnati all'Assistant per la correzione.

Conclusioni:

La funzionalità principale risponde alle aspettative di corretto funzionamento, non sono però stati implementati controlli che notifichino errori nei dati inseriti. Inoltre si può creare durante la fase di creazione di test una situazione d'errore che pregiudicate alcune funzionalità di correzione dell'Assistant.

Analisi dei Componenti

Questa sezione del documento si occupa di analizzare le informazioni contenute nel design document, ricercando in particolare i componenti principali che dovrebbero caratterizzare il progetto, e successivamente si verifica che questi componenti siano effettivamente presenti nell'applicazione svolgendo le funzionalità per cui sono stati ideati.

Seguendo l'ordine tenuto nel design document si procede all'analisi del data tier dell'applicazione.
In questa sezione viene descritta mediante un design concettuale ed un design logico la base di dati del sistema, queste descrizioni vengono realizzate rispettivamente mediante un diagramma ER ed un diagramma logico.
Analizzando il contenuto del documento e confrontandolo con ciò che è stato realmente implementato nella base di dati si nota che le tabelle definite nell'applicazione rispettano la forma definita nel design document. Si può tuttavia appuntare che nei diagrammi mancano alcune entità che sono invece state implementate nel database ed in particolare si fa riferimento alle tabelle che definiscono i template, di queste entità infatti non è stato tenuto conto in nessuno dei due diagrammi presentati.
Continuando con l'analisi si è potuto osservare che sono state mantenute le relazioni definite tra le varie entità e che tali relazioni sono state implementate in modo corretto seguendo le regole di traduzione delle relazioni in SQL.
Stesso discorso può essere fatto per quanto riguarda le gerarchie tra le entità, anche queste sono state rispettate e tradotte in modo corretto seguendo quanto era stato definito nel design document.

Successivamente si è passati all'osservazione della parte del documento dedicata al business tier. Questa è suddivisa in due sezioni, una in cui si descrivono gli entity beans ed una in cui si parla dei session bean.
E' necessario notare che queste sezioni sono state aggiornate nella guida di installazione dell'applicazione, per cui si fa riferimento a questo documento per valutare le parti che compongono il business tier.

Nella prima parte si parla appunto degli entity beans e viene riportato un diagramma in cui vengono indicati tutti gli entity che compongono il progetto e le loro relazioni.
Andando a confrontare quanto contenuto nel documento con ciò che invece è stato realmente implementato, si nota una totale coerenza per quanto riguarda le classi. Infatti tutti gli entity descritti nel diagramma sono stati tutti implementati come classi dell'applicazione. Anche per quel che riguarda le relazioni tra gli entity si può osservare che sono state implementate al meglio tutte le associazioni tenendo conto delle loro cardinalità. Infine si può dedurre che sia i metodi che gli attributi definiti dai diagrammi vengono implementati nelle rispettive classi.

La seconda parte si occupa come detto degli session bean, anche in questo caso si fa riferimento al diagramma contenuto nella guida di installazione in quanto risulta aggiornato rispetto a quello contenuto nel design document. Innanzitutto si può osservare che le classi presenti nello schema sono state definite anche all'interno del progetto mostrando quindi una totale coerenza. Inoltre i metodi definiti nello schema delle classi coincidono con quelli realmente implementati, l'unica eccezione degna di nota la si può trovare nella classe "Common" in cui i metodi implementati risultano in numero maggiore rispetto a quelli dichiarati nel documento.

Completata l'analisi della parte relativa al business tier, si descrive nel seguito la sezione relativa al presentation tier. Questo si occupa di fornire un'interfaccia tra l'applicazione e l'utente consentendo a quest'ultimo di agire sui dati del sistema trascurando i dettagli implementativi.

Il paragrafo del design document dedicato a questo tier è suddiviso in tre parti. La prima contiene uno User Experience Diagram (UX-Diagram) in cui si mostra l'interazione tra l'utente e l'interfaccia del sistema.
Entrando nel dettaglio di quanto contenuto nel documento si nota che sono stati realizzati due diagrammi, uno per descrivere i componenti relativi all'interazione del trainer col sistema e uno per l'interazione del trainee.
La fase di analisi di questa sezione consisteva nel confrontare i componenti dei diagrammi con le classi contenute nella sezione web del progetto, andando a verificare che tutti i componenti fossero stati implementati e che tutte le relazioni siano state rispettate.
Partendo con l'analisi delle interazioni tra il trainer e il sistema, si è visto che tutti gli <<*screen*>> e gli <<*input form*>> definiti nel diagramma sono presenti nel progetto ed ognuno di essi svolge le funzionalità per cui è stato ideato.

Si è riscontrato invece che alcune classi presenti nel progetto non sono state definite nel diagramma. Entrando nel dettaglio, queste classi sono relative alla creazione di relazioni tra Learning Object, alla creazione di Template ed infine alla creazione di un JUnitChecker.

Per quanto riguarda il diagramma relativo ad un trainee, dall'analisi si è potuto ottenere che anche in questo caso, tutto ciò che stato definito nel documento è presente nell'implementazione del progetto e svolge tutte le funzionalità per cui è stato ideato; ma anche in questo caso esistono delle classi che sono state implementate ma che non sono presenti nel diagramma, e sono quelle relative alla visualizzazione dei risultati ottenuti.

Il paragrafo del design document consta di due ulteriori parti, una in cui mediante dei diagrammi Boundary-Control-Entity vengono indicate le interazioni tra i componenti dell'applicazione ed una in cui utilizzando alcuni sequence diagram si descrivono alcuni casi d'uso del sistema. In tutti questi diagrammi sono stati descritti ed utilizzati i componenti già analizzati nelle precedenti sezioni del documento, per cui si può notare che è stata mantenuta una certa coerenza anche nelle fasi di descrizione delle interazioni tra i componenti.

Al termine dell'analisi del documento di design sono state individuate le incongruenze appena descritte, ma esiste un ulteriore incongruenza, o meglio una mancanza che non è relativa al design document ma è tuttavia importante. Infatti non è stata realizzata la documentazione del progetto, ovvero non è stato creato il JavaDoc delle classi. Questa assenza non permette di conoscere appieno le funzionalità del progetto, in quanto non viene specificato come è composta ogni singola classe e cosa realizza ogni suo metodo; per questo risulta complicato a chiunque voglia utilizzare l'applicazione intuire ciò che realizza ogni sua sezione senza dover analizzare direttamente il codice.
Bisogna comunque sottolineare che nei documenti allegati al progetto e nella guida all'installazione viene specificato il funzionamento di molte parti dell'applicazione e questi risultano utili ai fini della comprensione da parte di un utente qualsiasi. Resta però il fatto che la documentazione di un progetto è di fondamentale importanza e che si tratta di uno dei principali documenti a cui fare riferimento in caso di difficoltà nell'utilizzo dell'applicazione.