

POLITECNICO DI MILANO

Facoltà di Ingegneria dell'Informazione
Corso di Laurea in Ingegneria Informatica
Progetto di Ingegneria del Software 2



LESSON: LEarn Stuff Simply ONline
Quality Control and
Testing

Prof. Mirandola Raffaella
Prof. Ghezzi Carlo

Alessandro Sisto - 678319
Andrea Valdi - 713278
Massimo Vitiello - 704004

Anno Accademico 2010-2011

Indice

1	Introduzione	2
1.1	Note Preliminari	2
2	Controllo qualità	3
2.1	Analisi Generale	3
2.2	Black Box Testing	4
2.2.1	Note riguardo al testing	4
2.2.2	Testing <i>Login</i>	5
2.2.3	Testing <i>Registrazione</i>	5
2.2.4	Testing <i>Inserimento Corsi</i>	6
2.2.5	Testing <i>Inserimento materiale corso</i>	6
2.2.6	Testing <i>Aggiunta domande (risposta aperta, risposta multipla, esercizio)</i>	7
2.2.7	Testing <i>Aggiunta nuovo Test</i>	7
2.2.8	Testing <i>Aggiunta nuovo Template</i>	7
2.2.9	Testing <i>Gestione delle richieste di iscrizione</i>	7
3	Valutazione complessiva finale	9

Capitolo 1

Introduzione

L'obiettivo di questo elaborato è quello di documentare il controllo qualità effettuato sul progetto *se2project* reperibile via web all'indirizzo <http://se2project.origo.ethz.ch/download>.

Il seguito di questo documento è strutturato in 3 parti principali.

- Nella prima sezione vengono forniti i risultati di un'analisi generale del progetto.
- Nella seconda parte vengono riportate le analisi derivanti da operazioni di *Black Box Testing*. Non entriamo quindi nel dettaglio del codice sviluppato, limitandoci a controllare che l'applicazione sotto test includa tutto ciò che era previsto nelle specifiche iniziali.
- Nell'ultima parte viene effettuata una valutazione complessiva del sistema.

1.1 Note Preliminari

Le operazioni contenute in questo documento sono state eseguite dopo avere effettuato il deploy del progetto sia su JBoss 6.0 che su JBoss 5.1.

Capitolo 2

Controllo qualità

2.1 Analisi Generale

Il sistema risulta essere sviluppato in Java e la release finale viene fornita con manuale d'installazione e Javadoc delle funzioni principali. Il deploy su JBoss può essere effettuato facilmente seguendo i passi previsti nel manuale d'installazione. Il client EJB è rappresentato da un'applicazione Web, anch'essa facilmente 'installabile' (*deployable*) attraverso JBoss. Infine, il DBMS utilizzato è MySql.

Sono state rilevate diverse incongruenze tra le specifiche riportate all'interno del Design Document e quanto effettivamente implementato. In particolare la definizione di gran parte dei Session Bean, ipotizzata come Stateful nel DD, è in realtà implementata da soli Stateless Bean. Ciò vale anche per la classe *LoginSystem*, il cui scopo è quello di fornire l'autenticazione degli utenti al sistema. Questo porta a concludere che nell'implementazione finale si è scelto di mantenere la sessione di autenticazione dell'utente all'interno dell'applicazione client, invece di gestirla nel Business Tier.

Una scelta di questo tipo è facilmente opinabile, in quanto implica la presenza di un forte legame fiduciario tra Business Tier e Client Tier, non sempre ottenibile. Le interfacce implementate dai Session Bean sono inoltre di tipo 'Remote', ulteriore aggravante del problema citato. Un ipotetico sviluppatore potrebbe quindi implementare una semplice applicazione Client che, connettendosi ai servizi offerti dalle API del Business Tier, supererebbe ogni tipo di autenticazione, permettendo all'utente di manipolare il sistema in modo incontrollato.

Ciò è facilmente verificabile attraverso un semplice tentativo di connessione diretta a una pagina interna del sito, senza aver prima effettuato l'ope-

razione di login. Test di questo genere sono riportati nelle sezioni successive di questo documento.

Dal punto di vista del client, a prima vista l'applicazione web si presenta graficamente scarna e poco user-friendly. Le pagine web non includono una definizione degli stili CSS, e ciò potrebbe rendere la navigazione meno gradevole.

Ciò nonostante, le funzioni principali, se controllate all'interno di test-set 'normali' e non degeneri, eseguono nel complesso quasi tutte le operazioni richieste senza causare errori nel sistema.

In caso di errore, la maggior parte delle volte, viene ritornata un pagina che avvisa l'utente che si è verificato un errore, ma senza specificarne la causa. Di conseguenza non è immediato capire cosa abbia generato l'errore.

Per quanto riguarda la navigazione, l'unico landmark presente è quello che rimanda alla pagina principale dell'utente. Non sono quindi presenti il tasto di back o altri landmark che faciliterebbero la navigazione attraverso le pagine.

2.2 Black Box Testing

Nel seguito del testo verrà presentato un sottocapitolo per ogni macrofunzione controllata. Per ognuna di queste sezioni verrà fornito un elenco di note positive (segnalate con un +) e uno di note negative (segnalate con un -), in modo da documentare gli aspetti positivi e i difetti riscontrati durante l'esecuzione dei test.

Non sono stati eseguiti test con l'utilizzo di JUnit, in quanto buona parte dei controlli sui campi inseriti dagli utenti avvengono nel Web Tier e, in caso di fallimento, le Servlet non rispondono utilizzando i codici d'errore previsti dallo standard HTTP, rendendone vano il testing automatico con JUnit.

2.2.1 Note riguardo al testing

I risultati del controllo qualità riportati nel seguito derivano da operazioni di testing a scatola chiusa effettuato sulle *sole* operazioni di cui siamo effettivamente riusciti a controllare il funzionamento. Parte delle funzionalità previste e specificate nel manuale utente, non sono state controllate per via di errori generici e crash non chiaramente comprensibili rilevati durante i test, sebbene quest'ultimi siano stati eseguiti seguendo alla pari le istruzioni riportate nel manuale utente.

2.2.2 Testing *Login*

Specifici INPUT provati:

- username = '1', password = 'a'
- username = 'a', password = 'a'

Note:

- + Il controllo dei campi del login avviene sia lato client che lato server (Web Tier).
- Se si effettua il login come Trainee è possibile forzare il sistema (inserendo direttamente l'URL) in modo da vedere la pagina di Trainer, Assistant, etc... Vale anche il contrario.
- + Se però si tenta di eseguire un'operazione non assegnata al proprio ruolo (tipo Trainee inserisce nuovo corso), il sistema lo impedisce riportando un errore generico.
- È possibile accedere alle pagine interne del sito senza aver effettuato alcun login.

2.2.3 Testing *Registrazione*

Specifici INPUT provati:

- data = -1/-1/-2
- data = 01/01/2030
- data = 01/1/0000
- data = 01/01/-2030

Note:

- Nell'HTML non vi è un tag <form>, il submit avviene tramite una funzione Javascript che 'costruisce' la chiamata GET ed effettua un redirect. Questo implica che, se Javascript dovesse essere disattivato, non funzionerebbe.
- Non avviene il controllo della registrazione di un utente con lo stesso nome. (comunque nel login viene usato l'id, quindi la cosa non incide sul funzionamento dell'autenticazione utente).

- Non avviene un controllo lato server sulla data. E' possibile impostare una data del tipo inserendo i dati all'interno della chiamata GET, senza che venga riportato errore e senza che venga bloccata l'operazione (ovvero viene salvata nel DB una data inconsistente).
- I controlli circa le data di nascita non vengono effettuati in maniera adeguata. Viene infatti controllato solamente che il giorno sia un numero compreso tra 1 e 31, che il mese sia un numero compreso tra 1 e 12 e che l'anno sia un numero composto da non più di 4 cifre. Di conseguenza sono valide date quali 31/02/1987 o 01/01/5625. A video vengono richiesti giorno in formato gg, mese in formato mm e anno in formato aaaa. Inserendo però date come 1/1/2000 o 01/2/1999 queste vengono comunque accettate. Altra incorrettezza riguarda il campo anno, in cui è possibile inserire anche date negative.
- I controlli vengono effettuati solamente circa le date. E' infatti possibile inserire un utente lasciando tutti gli altri campi vuoti (inclusa la password).
- Se vengono inserite stringhe molto lunghe (10000 chars) il sistema crasha.
- Si possono usare simboli di punteggiatura / caratteri non ascii nei campi senza nessun errore riscontrato.

2.2.4 Testing *Inserimento Corsi*

Specifici INPUT provati:

- crediti = 'abc'

Note:

- È possibile inserire più corsi con lo stesso nome.
- Se si inserisce una stringa come numero di crediti (inserendo il parametro direttamente nella chiamata GET) non viene gestita l'eccezione *NumberFormatException*. La stessa cosa avviene se non si mette uno dei parametri richiesti (*NullPointerException*).

2.2.5 Testing *Inserimento materiale corso*

Note:

- Non è possibile inserire il materiale del corso. Una volta inserito il nome del materiale ed assegnatogli il corso, bisogna infatti assegnargli la priorità rispetto agli altri materiali del corso tramite click su apposito tasto. In questo punto però, il sistema crasha (*JasperException*).

2.2.6 Testing *Aggiunta domande (risposta aperta, risposta multipla, esercizio)*

Note:

- + L'inserimento delle domande nel sistema avviene correttamente.

2.2.7 Testing *Aggiunta nuovo Test*

Note:

- Per creare un test è necessario aver creato prima le domande nella sezione apposta e quindi l'inserimento risulta molto macchinoso e di non facile comprensione pur seguendo il manuale utente. Tuttavia, nella pagina di creazione del test, per ogni tipo di domanda viene fornita una lista di quelle disponibili per facilitare l'inserimento.
- Non è stato possibile provare il correttore JUnit, in quanto impossibile eseguire l'upload file per mancanza delle librerie necessarie.
- Non è possibile creare un test in quanto una volta compilati tutti i campi e cercando di inviare i dati il sistema và in errore.
- Se si cerca di creare un test partendo da un template, ma il template non è stato creato, il sistema và in errore (*JasperException*).

2.2.8 Testing *Aggiunta nuovo Template*

Note:

- Non è possibile creare un template in quanto una volta compilati tutti i campi e cercando di inviare i dati il sistema và in errore.

2.2.9 Testing *Gestione delle richieste di iscrizione*

Note:

- L'arrivo delle richieste di iscrizione al corso al professore non viene segnalato: l'unico modo che ha il professore per accorgersi delle notifiche è entrare nella sezione apposita.
- Dualmente, non viene notificata allo studente l'esito della richiesta: l'unico modo per accorgersi circa l'esito è quella di entrare nella sezione del materiale o del test, in quanto qui viene visualizzata la lista dei corsi a cui è iscritto.

Capitolo 3

Valutazione complessiva finale

L'applicazione sviluppata risulta nel complesso funzionante, ma di difficile utilizzo. Molte delle funzioni sono utilizzabili solo se si ha una precedente conoscenza della natura del progetto, difficilmente un utente finale potrebbe riuscire a sfruttarle pienamente.

Nota di merito è data dal tentativo di sviluppare la maggioranza delle funzionalità previste dai requisiti iniziali evitando il più possibile l'occorrenza di eventi tali da necessitare il riavvio dell'applicativo, sebbene spesso ciò comporti scelte implementative poco eleganti.

Molti degli errori riscontrati durante il controllo qualità non sono stati di facile comprensione, specialmente a causa della scarsa personalizzazione dei feedback d'errore restituiti all'utente.

Complessivamente, il sistema non risulta essere sviluppato nell'ottica di future espansioni (eventuali nuovi Application Client) ed è strutturato per poter funzionare in un limitato dominio d'esecuzione.