Progetto di Ingegneria del Software 2

SWIMv2: Small World Hypothesis Machine v2

Autori: Professore:   
Bulla Jacopo Mottola Luca  
Caio Davide  
Cappa Stefano

Questo documento rappresenta la quinta ed ultima deliverable.

Lo scopo è di eseguire il testing del software realizzato da un altro team di sviluppo. Ovviamente, non consiste solo nel verificare il corretto funzionamento del prodotto, ma anche di esaminare la documentazione individuando eventuali incongruenze.

Indice

1. Introduzione 4

2. Testing degli scenari 5

2.1. Scenario 1 5

2.2. Scenario 2 7

2.3. Scenario 3 9

2.4. Scenario 4 10

2.5. Scenario 5 13

2.6. Scenario 6 17

2.7. Scenario 7 21

2.8. Scenario 8 22

2.9. Scenario 9 23

3. Test aggiuntivi 24

Indice delle figure 29

# Introduzione

Questo documento rappresenta la fase di testing sul progetto (<http://code.google.com/p/bernasconi-calabrese-hessel/>) di un altro team di sviluppo.

Essa è composta da:

* Anna Bernasconi
* Chiara Calabrese
* Matteo Hessel

La fase di testing, descritta nel Project Planning 1.4, è stata sviluppata e suddivisa in nuove sotto-fasi.

1. Verifica del manuale d’installazione.
2. Ricerca di eventuali incongruenze nel RASD e nel DD.
3. Sviluppo di alcuni casi di test, specificandone il risultato.
4. Aggiunta di alcuni consigli verso il team di sviluppo, per migliorare il prodotto.

## Definizioni ed abbreviazioni

* Team di testing: gruppo composto da Bulla, Caio e Cappa
* Team di sviluppo: gruppo composta da Bernasconi, Calabrese e Hessel.
* Repository: è un ambiente di un sistema informativo (di tipo ERP), in cui sono gestiti i metadati, attraverso tabelle relazionali; l'insieme di tabelle, regole e motori di calcolo tramite cui si gestiscono i metadati prendono il nome di metabase. (fonte: wikipedia.org)
* Deliverable: sinonimo di “consegna”

# Installazione

**Il team di sviluppo non ha fornito il prodotto come file eseguibili** (.jar, .war oppure un .ear che li ingloba), ma solamente un collegamento http al “repository” SVN di Google Code.

Poiché nei documenti forniti come deliverable **non sono state fatte considerazioni in merito ai destinatari del prodotto**, si presume che essi abbiano le conoscenze necessarie per installarlo.

Nel caso il committente sia una società/gruppo esterno tale ipotesi potrebbe essere una forzatura e potenzialmente improbabile, poiché solitamente il committente richiede un lavoro ad un gruppo di sviluppatori, proprio perché non ha le conoscenze necessarie.

Nel caso in cui il prodotto sia sviluppato da studenti per un tutor o professore, allora tali ipotesi sono certamente valide, ma in ogni caso **sarebbe stato più corretto precisarlo.**

**Tutte le procedure descritte nel documento d’installazione sono molto generiche e poco approfondite.**

Nel caso in cui il team di sviluppo avesse fornito gli eseguibili, **le procedure legate ad Eclipse** sarebbero state secondarie. Però nel caso in esame **costituiscono l’intero manuale di installazione e per tale motivo sarebbero dovute esser descritte in modo molto più approfondito.**

Per esempio,nel capitolo 3 **la procedura d’installazione del plug-in SVN è poco dettagliata**.

Nel capitolo 4, quando è descritta la procedura di configurazione di MySQL, è presente la seguente frase: *“Inoltre per inserire il* ***connettore*** *abbiamo usato MySQL Workbench.* ***Dopo averla installata*** *abbiamo* ***selezionato New Connection e creato in questo modo la connessione chiamandola swim\_project****”*

A cosa si riferisce la parola “connettore”?

Il team di testing ha ipotizzato che si trattasse del file **“mysql-connector-java-*versione*-bin.jar”**. Però, in tal caso la frase citata non avrebbe senso, poiché questo file .jar **sarebbe importato direttamente nel server e non avrebbe nulla a che vedere con MySQL Workbench.**

Inoltre, anche **la frase successiva è errata, perché dice di creare la connessione come “swim\_project”,** quando in realtà quel nome non è rilevante. Invece, **è importante che il nome del database (“schema”) da creare sia “swim\_project”, poiché configurato nel file persistence.xml all’interno del progetto.**

**Si noti inoltre una grave mancanza: non è stato spiegato che, dopo aver creato la connessione al database, bisogna anche creare il nuovo “schema” (nome del database) tramite MySQL Workbench, chiamandolo “swim\_project”.**

**Nota bene: il prodotto è impostato in modo che ad ogni avvio generi un nuovo database, poiché il file “persistence.xml” contiene la voce “create-drop”. Sarebbe stato più corretto fornire il software con la voce “update” ed un file .sql contenente le query per generare la base di dati.**

Tra l’altro la generazione di queste query può essere facilmente realizzata tramite MySQL Workbench, scegliendo addirittura di inserire alcuni dati predefiniti.

Nel capitolo 5, il problema è che **il team di sviluppo ha realizzato il software tramite Java7 (1.7), nonostante JBoss AS 5.1 sia stato sviluppato per funzionare con la versione precedente o addirittura Java5 (http://anonsvn.jboss.org/repos/jbossas/tags/JBoss\_5\_1\_0\_GA/build/docs/readme.html).** Ovviamente, **tutto ciò ha richiesto la conversione dei progetti da Java7 a Java6**, applicando i seguenti cambiamenti prima per il progetto EJB e dopo per quello WEB:

1. Tasto destro sul progetto->”Properties”->”Java Compiler” e scegliere dal menu a tendina “1.6”, dopodiché salvare ed attendere che Eclipse aggiorni il progetto.
2. Tasto destro sul progetto->”Properties”->”Project Facets” e scegliere la versione “1.6” nella riga denominata “Java”.
3. Tasto destro sul progetto->”Properties”->”Java Build path” rimuovere JRE 1.7 ed importare la 1.6 tramite il pulsante “Add Library”->”JRE System Library”->”Workspace default JRE” e confermare l’operazione.

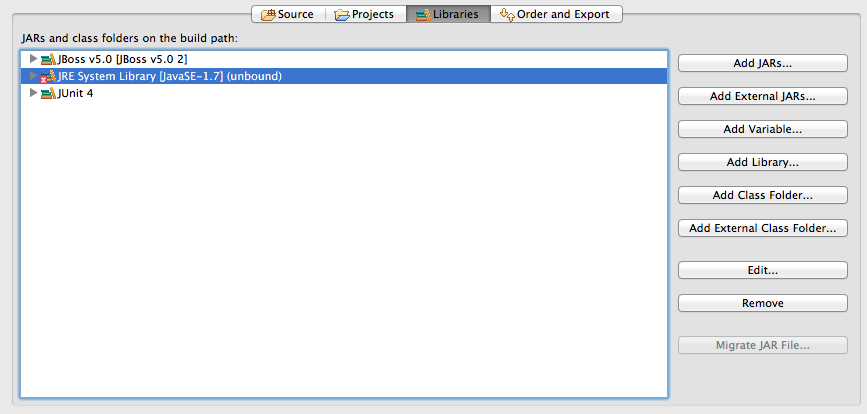


Fig. 2.1 - JRE 7.1 unbound

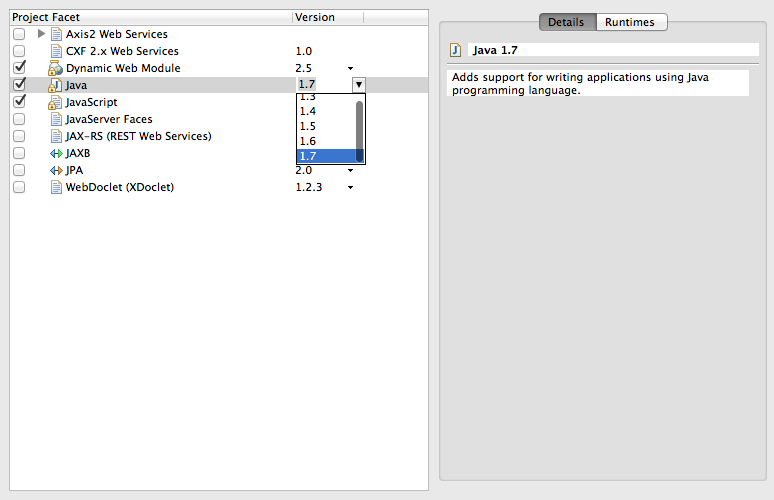


Fig. 2.2 – Problema Project Facets

E’ importante far notare che non è stato spiegato come impostare il server e cioè come aggiungerlo alle “Libraries” (Fig. 2.1).

**Inoltre, in Java6 il casting di un oggetto in (long) non è possibile, bensì è necessario utilizzare l’oggetto Long, cioè “(Long)”. Di conseguenza, il team di testing ha dovuto apportare tale modifica.**

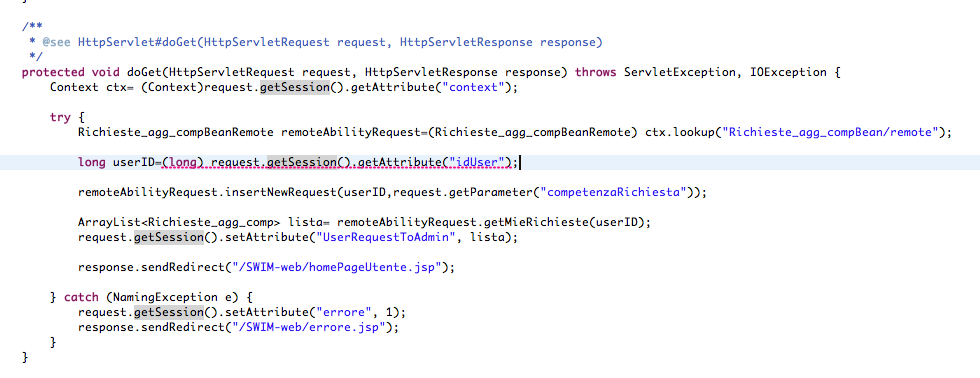


Fig. 2.3 - Problema di casting

**Per concludere, il team di testing ha messo in evidenza il fatto che i caratteri accentati nei nomi dei metodi e delle variabili, creano degli enormi problemi di funzionamento su piattaforme Mac OSX e Linux. Infatti, la codifica dei caratteri dei sistemi Microsoft è diversa e su Mac OSX appare così:**

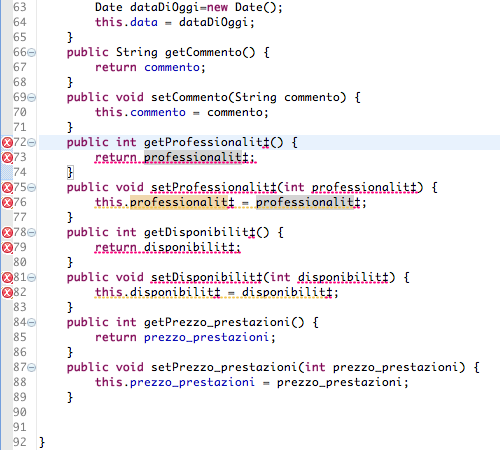


Fig. 2.4 - Problema codifica caratteri

Anche in questo caso un membro del team di testing ha dovuto sostituire manualmente tutti i caratteri errati con la “à” in tutti i file .java e .jsp (procedura tutt’altro che breve).

# Verifica della documentazione

# Casi di test

## Utente non registrato

1. Input: campi tutti vuoti

Output: nickname non inserito

Risultato: superato

1. Input: campi tutti vuoti

Output: nickname non inserito

Risultato: superato

1. Input: campi tutti vuoti

Output: nickname non inserito

Risultato: superato

1. Input: campi tutti vuoti

Output: nickname non inserito

Risultato: superato

1. Input: campi tutti vuoti

Output: nickname non inserito

Risultato: superato

1. Input: campi tutti vuoti

Output: nickname non inserito

Risultato: superato

## Utente registrato

* + 1. Rifiuto di una richiesta di aiuto

1. Input: pulsante **rifiuta**

Output: nessuno

Risultato: superato

* + 1. Inserisci una valutazione

1. Input: campi punteggio default, campi testo vuoti

Output: valutazione inserita

Risultato: superato

Commento: nella pagina dell’utente valutato, tra i commenti compare la scritta “Commento:” anche se non è stato lasciato alcun commento

1. Input: commento di lunghezza normale, punteggio

Output: valutazione inserita

Risultato: superato

Commento: vedi sezione consigli

1. Input: commento lungo, punteggio

Output: Eccezione non gestita

Risultato: fallito

Commento: sarebbe stato meglio gestire una lunghezza massima nelle aree di inserimento testo

* + 1. Logout

1. Input: nessuno

Output: pagina di login

Risultato: superato

## Amministratore

* + 1. Login

1. Input: nessuno

Output: login non effettuato

Risultato: superato

1. Input: nickname

Output: login non effettuato

Risultato: superato

1. Input: password

Output: login non effettuato

Risultato: superato

1. Input: nickname, password

Output: login effettuato

Risultato: superato

* + 1. Gestione richieste utente

1. Input: esiste già

Output: abilità non aggiunta

Risultato: superato

1. Input: rifiuta

Output: abilità non aggiunta

Risultato: superato

1. Input: nessuno

Output: abilità non aggiunta

Risultato: fallito

1. Input: codice competenza

Output: abilità non aggiunta

Risultato: fallito

1. Input: competenza

Output: abilità non aggiunta

Risultato: fallito

1. Input: codice competenza, competenza

Output: abilità aggiunta

Risultato: superato

1. Input: codice competenza già esistente o competenza già esistente

Output: abilità non aggiunta

Risultato: superato

* + 1. Gestione archivio competenze

1. Input: competenza

Output: abilità non aggiunta

Risultato: fallito. Si può senza problemi aggiungere una competenza con uno o tutti e due i campi vuoti.

1. Input: codice competenza

Output: abilità non aggiunta

Risultato: fallito

1. Input: nessuno

Output: abilità non aggiunta

Risultato: fallito

1. Input: competenza, codice competenza

Output: abilità aggiunta

Risultato: superato

1. Input: competenza già aggiunta o codice competenza già aggiunto

Output: abilità non aggiunta

Risultato: superato

* + 1. Gestione utenti

1. Input: visualizza profilo

Output: pagina profilo

Risultato: superato

# Consigli aggiuntivi

## Pagina profilo utente

La sezione commenti dei feedback diventa illeggibile, una volta che iniziano ad esserci diversi commenti. Tutti i commenti sono attaccati senza spaziatura da un paragrafo all’altro, e non c’è nessun tipo di formattazione nel testo che faccia capire dove finisce un commento e ne inizia un altro.

# Indice delle figure

Fig. 2.1 - JRE 7.1 unbound 6

Fig. 2.2 – Problema Project Facets 6

Fig. 2.3 - Problema di casting 7

Fig. 2.4 - Problema codifica caratteri 7