

Test Report - TecStore

Natale Guadagno, Paolo Patrone

February 21, 2022

Contenuti

1	Metodologie di testing	4
1.1	JUnit	4
1.2	Selenium	6

Partecipanti

Nome	Matricola
Guadagno Natale	0512106546
Patrone Paolo	0512106153

Revision History

Data	Versione	Descrizione
21/2/2022	0.1	Prima stesura

1 Metodologie di testing

Si è scelto di dividere il testing in due fasi principali:

1.1 JUnit

Una volta raggiunta un'implementazione completa, ci si è adoperati per scrivere dei test JUnit che mettessero alla prova le funzioni principali del model, in maniera tale da rilevare eventuali errori nel codice o nell'esecuzione di particolari operazioni.

Si sono usati gli strumenti messi a disposizione da Eclipse IDE for Java EE per scrivere ed eseguire i test.

Da questo testing sono emerse svariate problematiche legate alla logica interna di alcune funzioni, dei Bean e alla struttura del database, che sono state prontamente risolte.


























































Poiché i test sono eseguiti sequenzialmente e un reset totale del database tra un test e l'altro rallenterebbe enormemente la fase di esecuzione dei test, è stato necessario rinominare alcuni dei test aggiungendo dei suffissi affinché, rispettando l'ordine alfabetico, l'ordine delle operazioni sia sempre possibile: ad esempio, se si tentasse di rimuovere un articolo o di modificarne le informazioni prima che venga inserito, ci sarebbero ovviamente degli errori.

Finished after 8.75 seconds

Runs: 52/52

Errors: 0

Failures: 0

- ▼  TestOrdineDAO [Runner: JUnit 5] (0.531 s)
 -  dettagliOrdine() (0.091 s)
 -  creazioneOrdineEmpty() (0.133 s)
 -  creazioneOrdine() (0.190 s)
 -  yyyElencoOrdiniMagazziniere() (0.007 s)
 -  wwwElencoRimborsi() (0.108 s)
- ▼  TestTicketDAO [Runner: JUnit 5] (0.387 s)
 -  dettagliTicket() (0.066 s)
 -  cambiaStatoNot() (0.014 s)
 -  elencoTicketCentralinistaNot() (0.020 s)
 -  elencoTicketCliente() (0.011 s)
 -  elencoTicketCentralinista() (0.009 s)
 -  rispostaTicketNot() (0.045 s)
 -  elencoMessaggiTicket() (0.018 s)
 -  rispostaTicket() (0.098 s)
 -  elencoTicketClienteNot() (0.005 s)
 -  elencoMessaggiTicketNot() (0.045 s)
 -  cambiaStato() (0.043 s)
 -  dettagliTicketNot() (0.012 s)
- ▼  TestArticoloDAO [Runner: JUnit 5] (0.401 s)
 -  modificaArticolo() (0.098 s)
 -  inserimentoNuovoArticolo() (0.051 s)
 -  ricercaArticolo() (0.021 s)
 -  ricercaArticoloNot() (0.006 s)
 -  elencoVenditeTipologia() (0.024 s)
 -  elencoVenditeTipologiaNot() (0.015 s)
 -  elencoVenditeCF() (0.018 s)
 -  modificaArticoloNot() (0.040 s)
 -  inserimentoNuovoArticoloNot() (0.000 s)
 -  elencoVenditeCentralinista() (0.007 s)
 -  zzzRimozioneArticolo() (0.083 s)
 -  elencoVenditeCFNot() (0.033 s)
- ▼  TestUtenteDAO [Runner: JUnit 5] (0.528 s)
 -  getTipologiaNot() (0.005 s)
 -  dettagliUtente() (0.006 s)
 -  modificaUtente() (0.104 s)
 -  existsNot() (0.006 s)
 -  ricercaDipendenti() (0.009 s)
 -  getTipologia() (0.005 s)
 -  exists() (0.005 s)
 -  autenticazione() (0.065 s)
 -  zEliminaDipendente() (0.025 s)
 -  dettagliUtenteNot() (0.025 s)
 -  autenticazioneNoEmail() (0.000 s)
 -  ricercaDipendentiNot() (0.012 s)
 -  modificaPasswordNot() (0.019 s)
 -  modificaPassword() (0.165 s)
 -  modificaUtenteNot() (0.033 s)
 -  autenticazioneNoPassword() (0.000 s)
 -  zEliminaUtente() (0.039 s)
- ▼  TestCarrelloDAO [Runner: JUnit 5] (0.223 s)
 -  aggiuntaArticoloNot() (0.009 s)
 -  aggiornamentoQuantitaCarrelloNot() (0.010 s)
 -  aggiornamentoQuantitaCarrello() (0.079 s)
 -  rimozioneArticolo() (0.043 s)
 -  aggiuntaArticolo() (0.030 s)
 -  getCarrello() (0.054 s)

1.2 Selenium

In seguito al testing JUnit e alle dovute modifiche al sistema, accertatisi della funzionalità quasi certa dei DAO si è usata la libreria Selenium per testare le servlet e le loro risposte.

Si è usato lo strumento Selenium IDE, un'estensione per browser che ha permesso di velocizzare enormemente la fase di scrittura dei test, riducendo al minimo la necessità di scrivere ulteriore codice.

Anche in questo caso sono emerse svariate criticità nell'esecuzione delle operazioni, che sono state risolte anche tramite la stesura dei test, che hanno velocizzato la fase di verifica del nuovo codice.

Come per i test JUnit, la sequenzialità delle operazioni è stato un fattore di cui si è tenuto conto nella stesura dei test, affinché, ammesso di partire da un database in cui esistono solo dei dati predefiniti, i test non falliscano a causa di dati estranei o imprevisti.