



Università degli Studi di Salerno

Corso di Ingegneria del Software

Test Execution Report



ZyphyK Sport

Docente:

Prof. Andrea De Lucia

Partecipanti:

Consiglio Luigi

D'Antuono Francesco Paolo

Tutor:

Gilberto Recupito

Vitale Ciro

Partecipanti al progetto:

nome	matricola	e-mail
Consiglio Luigi (CL)	0512110485	l.consiglio1@studenti.unisa.it
D'Antuono Francesco Paolo (DFP)	0512109798	f.dantuono10@studenti.unisa.it
Vitale Ciro (VC)	0512110719	c.vitale55@studenti.unisa.it

Revision History

data	versione	descrizione	autore
18/01/2023	1.00	inizio stesura introduction	DFP
18/01/2023	1.01	stesura introduction and relationship to other documents	DFP
18/01/2023	1.02	stesura test item transmittal report	CL
19/01/2023	1.03	stesura test log: unit testing	VC
22/01/2023	1.04	stesura test log: integration testing	VC
23/01/2023	1.05	revisione test log: system testing	CL
24/01/2023	1.06	revisione test log: unit e system testing	DFP,CL
25/01/2023	1.07	stesura test incident report: unit e integration testing	DFP
25/01/2023	1.08	stesura test incident report: system testing	CL,VC
27/01/2023	2.00	revisione del documento	Tutto il team

Test Execution Report	5
Introduction	5
Relationship to other documents	5
Test item transmittal report	5
Test log	6
Unit testing	6
TC_GestioneUtente	6
TC_GestioneUtenteDAO	6
Failure testDoUpdateClienti()	6
TC_GestioneUtenteServlet	6
TC_GestioneVendite	8
TC_GestioneVenditeDAO	8
Failure testDoRetrieveAllCartsContainsProds()	8
TC_GestioneVenditeServlet	8
Failure testDoGetAddToCart()	8
TC_GestioneCatalogo	10
TC_GestioneCatalogoDAO	10
Failure testDoSaveProducts()	10
TC_GestioneCatalogoServlet	11
Failure testDoPostAddInCatalog()	11
Integration testing	12
TC_GestioneUtente	12
TC_FR09-RegistrazioneIT	12
TC_FR10-AutenticazioneIT	12
TC_GestioneVendite	12
TC_FR08-AcquistoIT	12
TC_FR02-AggiornaStatoOrdineIT	12
TC_GestioneCatalogo	13
TC_FR01-InserimentoProdottoInCatalogoIT	13
TC_FR01-ModificaProdottoInCatalogoIT	13
System testing	14
TC_GestioneUtente	14
TC_FR09-Registrazione	14
TC_FR10-Autenticazione	17
TC_GestioneVendite	19
TC_FR08-Acquisto	19
TC_FR02-AggiornaStatoOrdine	22
TC_GestioneCatalogo	23
TC_FR01-InserimentoProdottoInCatalogo	23
TC_FR01-ModificaProdottoInCatalogo	25
Test Incident Report	27
Unit testing	27
Integration testing	28
System testing	28

Test Execution Report

Introduction

In questo documento andremo a trattare in modo dettagliato l'esecuzione dei test case e dunque notare le differenze tra il comportamento atteso e il comportamento osservato, evidenziando le eventuali anomalie riscontrate. Un test dove i risultati attesi rispecchiano i risultati osservati sarà definito *Passed* altrimenti sarà definito *Failed*.

Relationship to other documents

Il documento che stiamo realizzando ha una stretta relazione con i documenti di testing creati in precedenza. I documenti a cui si fa riferimento per il Test Execution Report sono:

- Test Plan: rappresenta la pianificazione del testing, dunque i modi e le strategie applicate per svolgerlo;
- Test Case Specification: rappresenta la specifica dei casi di test.

Test item transmittal report

Per eseguire i test, bisognerà andare nel package `src/test/java` del progetto. Da lì saranno suddivisi, in ulteriori package, tutti i testing di unità ed integrazione che sono stati effettuati, che permetteranno di distinguere in modo pratico le varie componenti. Per quanto riguarda l'esecuzione, basterà effettuare il run con JUnit delle varie classi di test. Inoltre, le classi si troveranno in uno stato in cui il run darà esecuzione corretta, cioè senza errori di compilazione o errori di utilizzo di classi e mock. Per testare le classi con i mock vi è la necessità di effettuare il run del codice con JUnit 4.

Test log

Qui sono mostrati tutti i confronti tra output attesi e output osservati e quindi tutti i log di test effettuati, che ci hanno permesso di riconoscere i failure nelle rispettive classi. Il testing di queste componenti è stato effettuato con JUnit e Mockito, che ci hanno permesso di testare componenti complesse in modo semplice e di riuscire a visualizzare le effettive failure che poi sono state corrette.

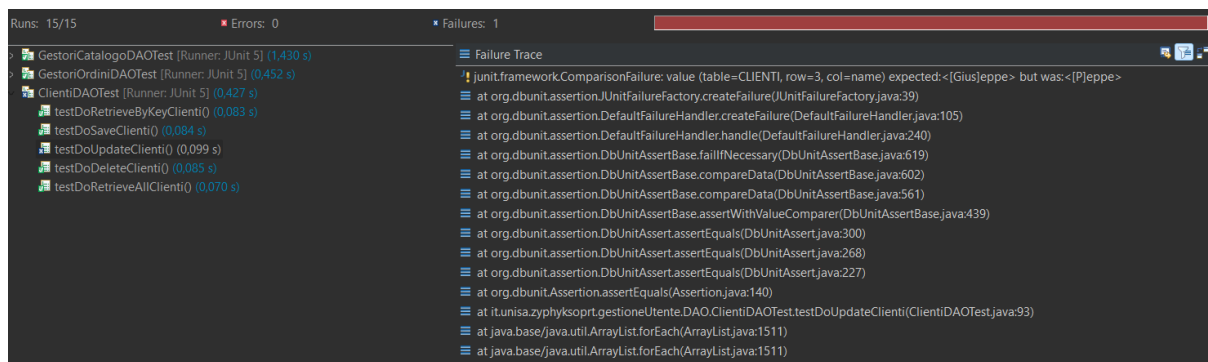
Unit testing

TC_GestioneUtente

In questo package sono raggruppati tutti i DAO e le Servlet inerenti al sottosistema GestioneUtente. Di seguito riportiamo le esecuzioni di test con i vari failure ritrovati.

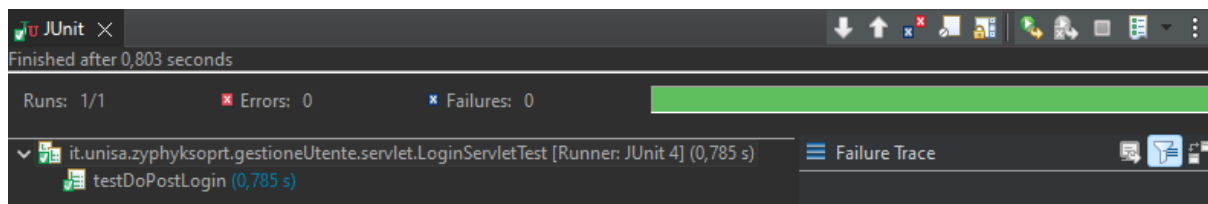
TC_GestioneUtenteDAO

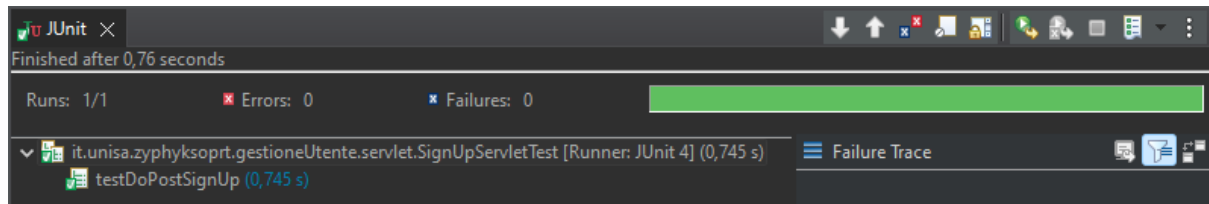
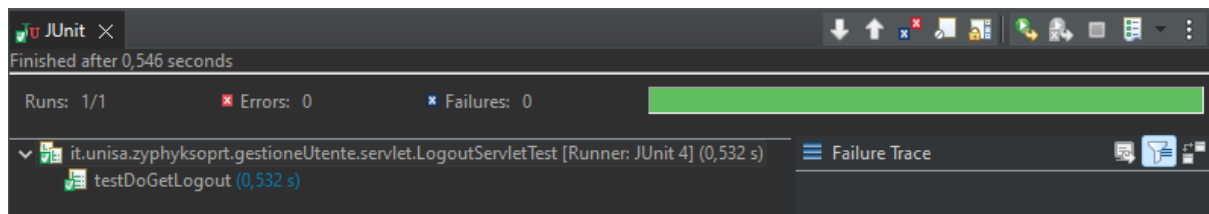
Failure testDoUpdateClienti()



In questo log viene riscontrata una failure della `doUpdate()` di `ClientiDAO`, che riguarda la differenza tra output atteso e osservato dell'aggiornamento dei campi di una tupla nel database.

TC_GestioneUtenteServlet

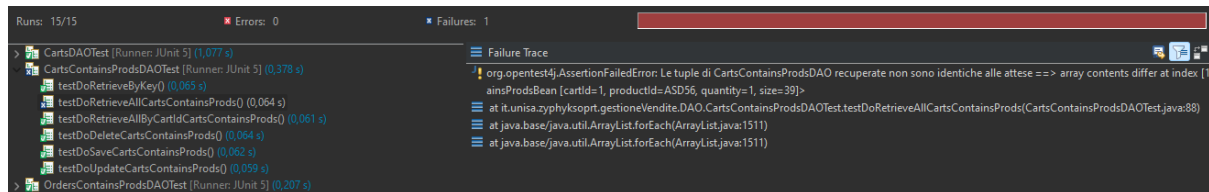




TC_GestioneVendite

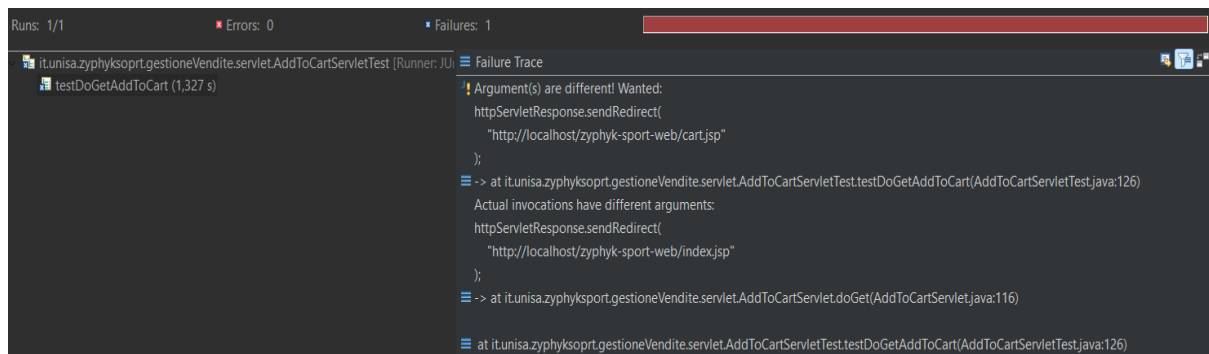
In questo package sono raggruppati tutti i DAO e le Servlet inerenti al sottosistema GestioneVendite. Di seguito riportiamo le esecuzioni di test con i vari failure ritrovati.

TC_GestioneVenditeDAO

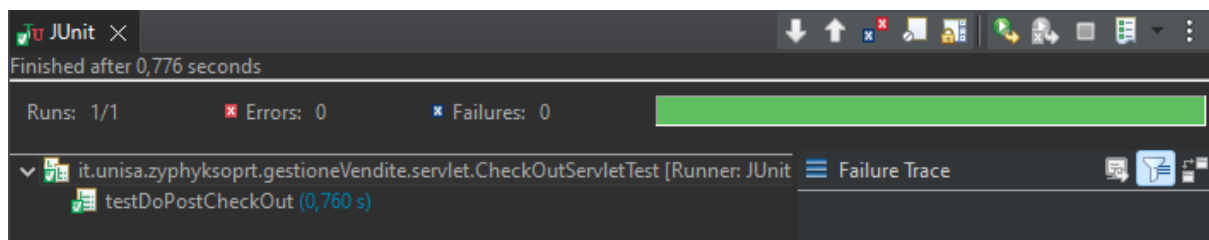
Failure testDoRetrieveAllCartsContainsProds()

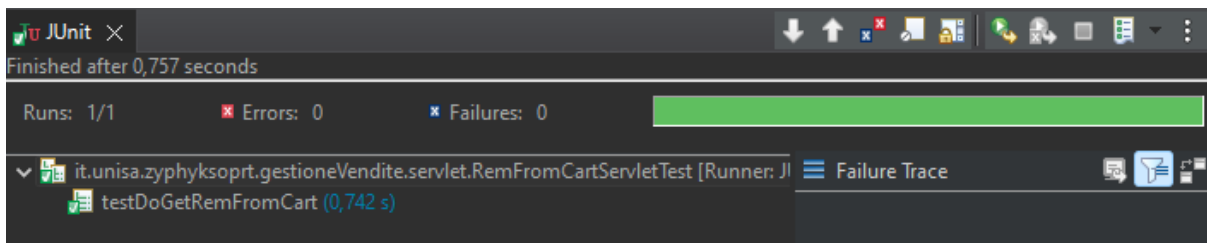
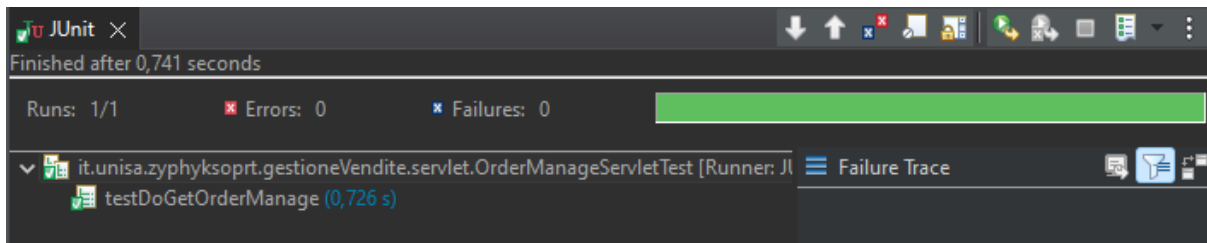
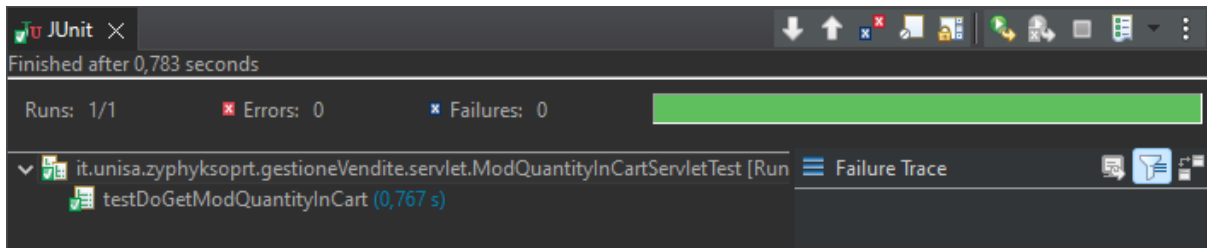
In questo log viene mostrato una failure nella *doRetrieveAll()* di *CartsContainsProdsDAO*, dove il confronto tra valore atteso e valore osservato di tutte le tuple nel database non corrispondono.

TC_GestioneVenditeServlet

Failure testDoGetAddToCart()

In questo log viene mostrato un failure nell'esecuzione del metodo *doGet()* della servlet *AddToCartServlet* poiché il campo *sendRedirect* della response attesa non corrisponde alla response osservata.



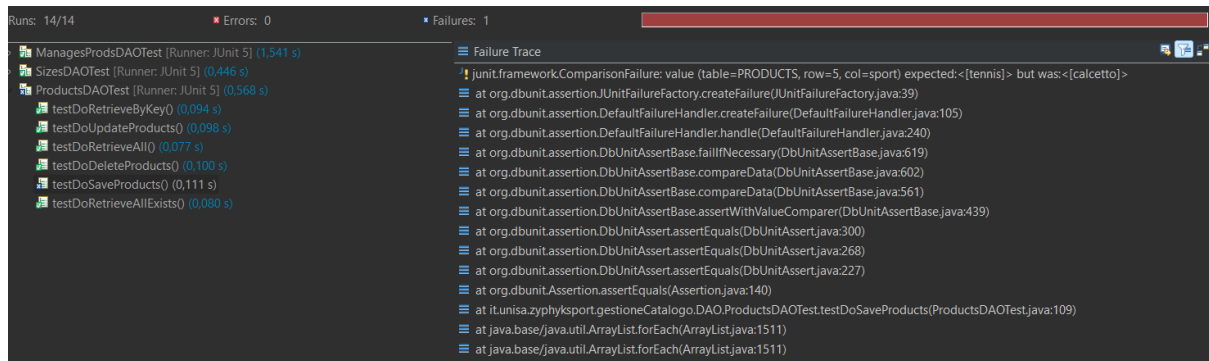


TC_GestioneCatalogo

In questo package sono raggruppati tutti i DAO e le Servlet inerenti al sottosistema Gestione, di seguito riportiamo le esecuzioni di test con i vari failure ritrovati.

TC_GestioneCatalogoDAO

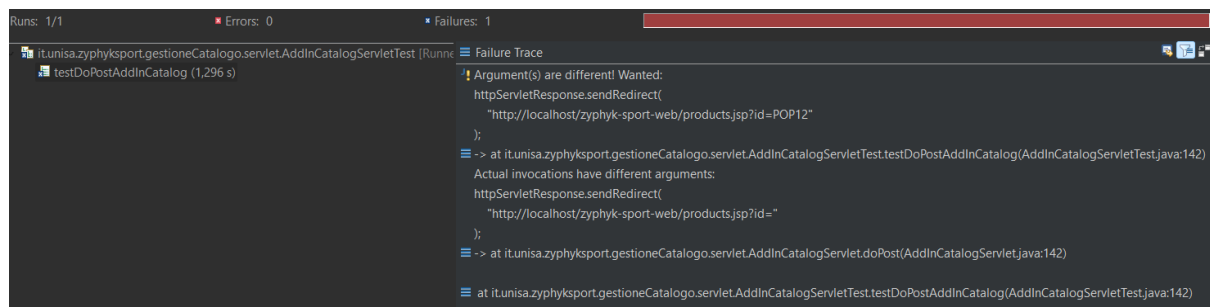
Failure testDoSaveProducts()



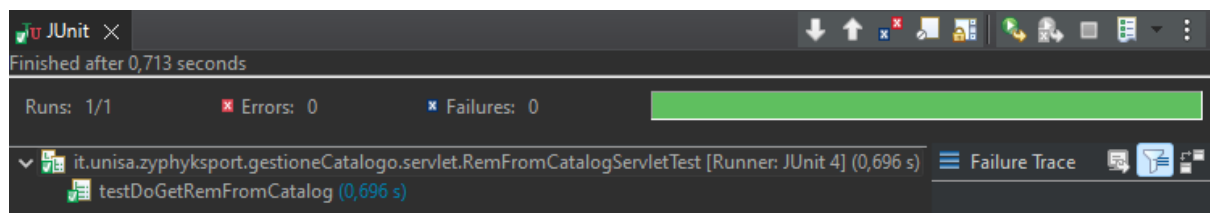
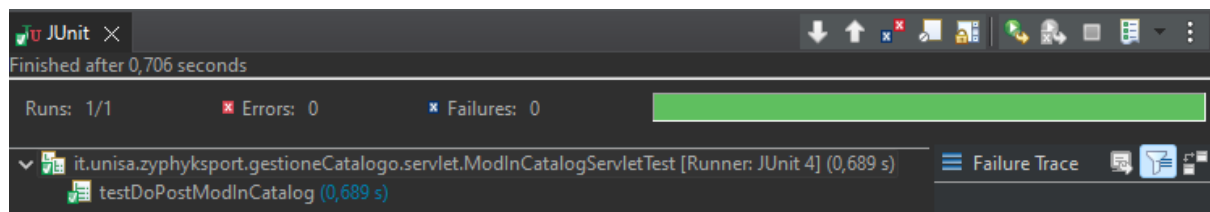
In questo log viene mostrato una failure nella *doSaveProducts()* in ProductsDAO dove il salvataggio di un nuovo prodotto nel database non viene effettuato in modo corretto poiché il valore atteso dello sport in cui si utilizza la scarpa, non corrisponde con il valore effettivo inviato.

TC_GestioneCatalogoServlet

Failure testDoPostAddInCatalog()



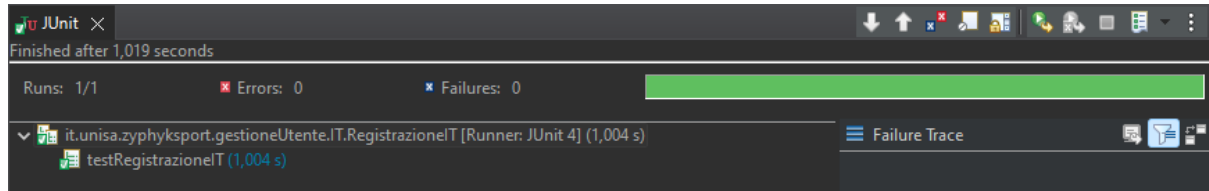
In questo log viene mostrato un failure nella `doPost()` `AddInCatalogServlet` dove la redirect del sito viene fatta in modo errato dunque la pagina visualizzata non corrisponde a quella attesa.



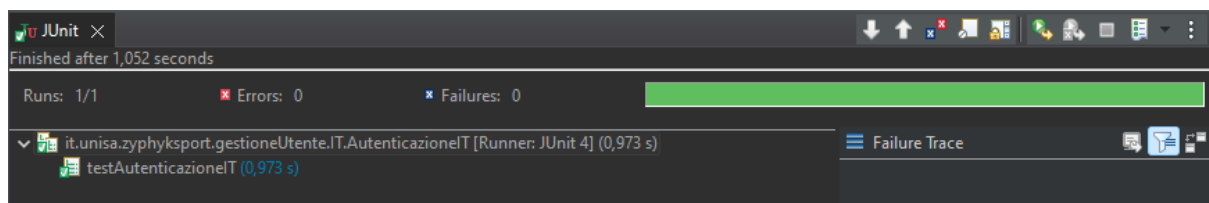
Integration testing

TC_GestioneUtente

TC_FR09-RegistrazioneIT

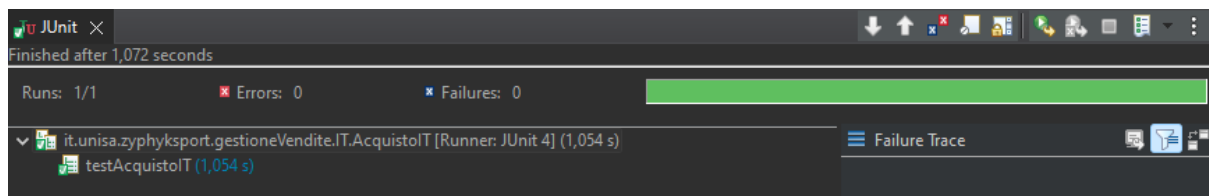


TC_FR10-AutenticazioneIT

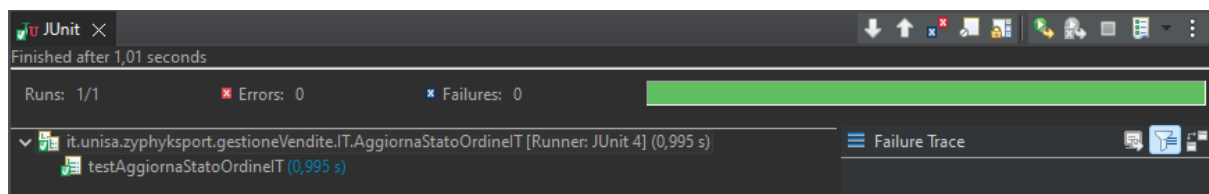


TC_GestioneVendite

TC_FR08-AcquistoIT

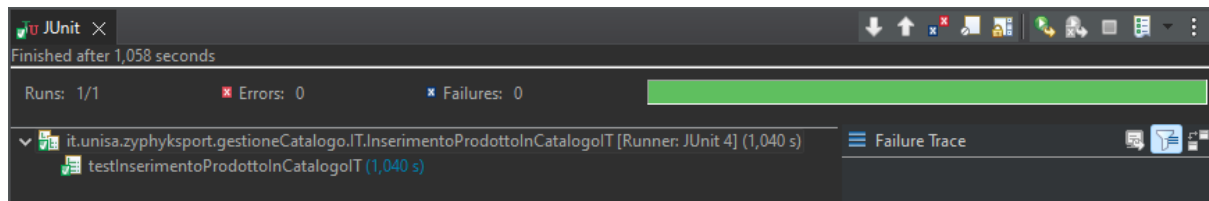


TC_FR02-AggiornaStatoOrdineIT

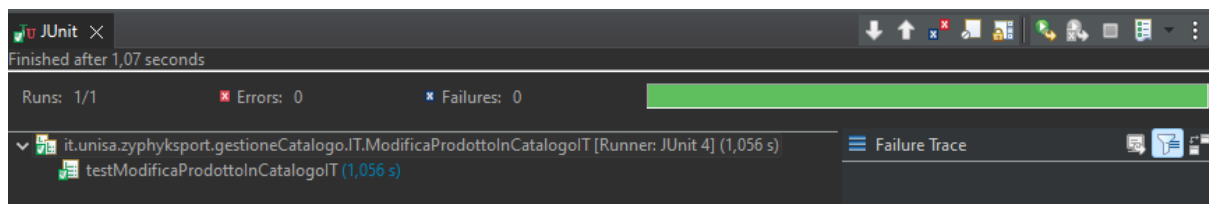


TC_GestioneCatalogo

TC_FR01-InserimentoProdottoInCatalogoIT



TC_FR01-ModificaProdottoInCatalogoIT



System testing

TC_GestioneUtente

TC_FR09-Registrazione

Test Case ID	TC_FR09-Registrazione_01
Output Atteso	Username non valido
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR09-Registrazione_02
Output Atteso	Username non valido
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR09-Registrazione_03
Output Atteso	Username già esistente
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR09-Registrazione_04
Output Atteso	Password non valida oppure non coincide con 'conferma password'
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR09-Registrazione_05
Output Atteso	Password non valida oppure non coincide con 'conferma password'
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR09-Registrazione_06
Output Atteso	Conferma password non valida
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR09-Registrazione_07
Output Atteso	Conferma password non valida
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR09-Registrazione_08
Output Atteso	Password non valida oppure non coincide con 'conferma password'
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR09-Registrazione_09
Output Atteso	Nome non valido
Anomalie	Nessuna anomalia
Data del Report	25/01/2023

Risultato	Passed
-----------	--------

Test Case ID	TC_FR09-Registrazione_10
Output Atteso	Nome non valido
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR09-Registrazione_11
Output Atteso	Cognome non valido
Anomalie	La lunghezza del cognome non valido non ha bloccato la registrazione
Data del Report	25/01/2023
Soluzione	Cambiare il controllo nella funzione javascript che effettua la verifica sulla lunghezza del cognome
Risultato	Failed

Test Case ID	TC_FR09-Registrazione_12
Output Atteso	Cognome non valido
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR09-Registrazione_13
Output Atteso	Data di nascita non valida
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR09-Registrazione_14
Output Atteso	Data di nascita non valida
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR09-Registrazione_15
Output Atteso	Email non valida
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR09-Registrazione_16
Output Atteso	Email non valida
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR09-Registrazione_17
Output Atteso	la registrazione è andata a buon fine
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

TC_FR10-Autenticazione

Test Case ID	TC_FR09-Autenticazione_01
Output Atteso	username non registrato o password errata
Anomalie	Nessuna anomalia
Data del Report	25/01/2023

Risultato	Passed
-----------	--------

Test Case ID	TC_FR09-Autenticazione_02
Output Atteso	username non registrato o password errata
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR09-Autenticazione_03
Output Atteso	il login viene effettuato
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

TC_GestioneVendite

TC_FR08-Acquisto

Test Case ID	TC_FR08-Acquisto_01
Output Atteso	via non valida
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR08-Acquisto_02
Output Atteso	via non valida
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR08-Acquisto_03
Output Atteso	numero civico non valido
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR08-Acquisto_04
Output Atteso	numero civico non valido
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR08-Acquisto_05
Output Atteso	città non valida
Anomalie	Nessuna anomalia

Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR08-Acquisto_06
Output Atteso	città non valida
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR08-Acquisto_07
Output Atteso	provincia non valida
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR08-Acquisto_08
Output Atteso	provincia non valida
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR08-Acquisto_09
Output Atteso	titolare carta non valido
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR08-Acquisto_10
Output Atteso	titolare carta non valido

Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR08-Acquisto_11
Output Atteso	numero carta non valido
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR08-Acquisto_12
Output Atteso	numero carta non valido
Anomalie	Il formato del numero di carta errato non ha bloccato l'acquisto
Data del Report	25/01/2023
Soluzione	Cambiare il controllo nella funzione javascript che effettua la verifica sul formato del numero di carta
Risultato	Failed

Test Case ID	TC_FR08-Acquisto_13
Output Atteso	cvv non valido
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR08-Acquisto_14
Output Atteso	cvv non valido
Anomalie	Nessuna anomalia
Data del Report	25/01/2023

Risultato	Passed
-----------	--------

Test Case ID	TC_FR08-Acquisto_15
Output Atteso	data scadenza carta non valida
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR08-Acquisto_16
Output Atteso	data scadenza carta non valida
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR08-Acquisto_17
Output Atteso	data scadenza carta non valida
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR08-Acquisto_18
Output Atteso	il cliente procede con l'acquisto e l'ordine viene creato
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

TC_FR02-AggiornaStatoOrdine

Test Case ID	TC_FR02-AggiornaStatoOrdine_01
Output Atteso	ordine già spedito

Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR02-AggiornaStatoOrdine_02
Output Atteso	l'aggiornamento dello stato viene fatto correttamente
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

TC_GestioneCatalogo

TC_FR01-InserimentoProdottoInCatalogo

Test Case ID	TC_FR01-InserimentoProdottoInCatalogo_01
Output Atteso	codice scarpa non valido
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR01-InserimentoProdottoInCatalogo_02
Output Atteso	codice scarpa non valido
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR01-InserimentoProdottoInCatalogo_03
Output Atteso	codice Scarpa già esistente
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR01-InserimentoProdottoInCatalogo_04
Output Atteso	Nome scarpa non valido
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR01-InserimentoProdottoInCatalogo_05
Output Atteso	Nome scarpa non valido
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR01-InserimentoProdottoInCatalogo_06
Output Atteso	Selezionare almeno una taglia da inserire
Anomalie	La taglia non è stata selezionata ma il prodotto è stato inserito
Data del Report	25/01/2023
Soluzione	Cambiare il controllo nella funzione javascript che effettua la verifica sulle taglie selezionate: almeno una per inserire il prodotto
Risultato	Failed

Test Case ID	TC_FR01-InserimentoProdottoInCatalogo_07
Output Atteso	Costo non valido
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR01-InserimentoProdottoInCatalogo_08
--------------	--

Output Atteso	Costo non valido
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR01-InserimentoProdottoInCatalogo_09
Output Atteso	il gestore visualizza il catalogo con il prodotto appena inserito
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

TC_FR01-ModificaProdottoInCatalogo

Test Case ID	TC_FR01-ModificaProdottoInCatalogo_01
Output Atteso	prodotto non esistente
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR01-ModificaProdottoInCatalogo_02
Output Atteso	nome scarpa non valido
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR01-ModificaProdottoInCatalogo_03
Output Atteso	nome scarpa non valido
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR01-ModificaProdottoInCatalogo_04
Output Atteso	selezionare almeno una taglia da inserire
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR01-ModificaProdottoInCatalogo_05
Output Atteso	costo non valido
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR01-ModificaProdottoInCatalogo_06
Output Atteso	costo non valido
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Case ID	TC_FR01-ModificaProdottoInCatalogo_07
Output Atteso	il gestore visualizza il catalogo con il prodotto appena inserito
Anomalie	Nessuna anomalia
Data del Report	25/01/2023
Risultato	Passed

Test Incident Report

Dopo aver confrontato i risultati previsti con quelli ottenuti, abbiamo identificato diversi failure che hanno causato incident nel sistema. Di seguito elenchiamo questi failure e i relativi incident, che ricordiamo sono le conseguenze dei failure stessi.

Unit testing

I failure riscontrati sono dovuti soprattutto da errori di query nei DAO e redirect errate nelle Servlet. Inoltre abbiamo applicato anche Equivalence Class Testing per testare gli output attesi con gli output osservati.

Le failure riscontrate sono:

- *Failure testDoUpdateClienti()*: l'incident è il non aggiornamento della tupla, si procederà alla revisione del metodo *doUpdate()* di *ClientiDAO* che ha portato alla failure con conseguente riesecuzione del test case, di seguito il failure *testDoUpdateClienti()*:

```

Runs: 15/15      Errors: 0      Failures: 1
ClientiDAOTest [Runner: JUnit 5] (1.430 s)
  testDoRetrieveByKeyClienti() (0.083 s)
  testDoSaveClienti() (0.084 s)
  testDoUpdateClienti() (0.099 s)
  testDoDeleteClienti() (0.085 s)
  testDoRetrieveAllClienti() (0.070 s)
Failure Trace
  junit.framework.ComparisonFailure: value (table=CLIENTI, row=3, col=name) expected: <[Gius]eppe> but was: <[P]eppe>
    at org.dbunit.assertion.JUnitFailureFactory.createFailure(JUnitFailureFactory.java:39)
    at org.dbunit.assertion.DefaultFailureHandler.createFailure(DefaultFailureHandler.java:105)
    at org.dbunit.assertion.DefaultFailureHandler.handle(DefaultFailureHandler.java:240)
    at org.dbunit.assertion.DbUnitAssertBase.failIfNecessary(DbUnitAssertBase.java:619)
    at org.dbunit.assertion.DbUnitAssertBase.compareData(DbUnitAssertBase.java:602)
    at org.dbunit.assertion.DbUnitAssertBase.compareData(DbUnitAssertBase.java:561)
    at org.dbunit.assertion.DbUnitAssertBase.assertWithValueComparator(DbUnitAssertBase.java:439)
    at org.dbunit.assertion.DbUnitAssert.assertEquals(DbUnitAssert.java:300)
    at org.dbunit.assertion.DbUnitAssert.assertEquals(DbUnitAssert.java:268)
    at org.dbunit.assertion.DbUnitAssert.assertEquals(DbUnitAssert.java:227)
    at org.dbunit.assertion.Assertion.assertEquals(Assertion.java:140)
    at it.unisa.zyphysoprt.gestioneUtente.DAO.ClientiDAOTest.testDoUpdateClienti(ClientiDAOTest.java:93)
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
  
```

- *Failure testDoRetrieveAllCartsContainsProds()*: l'incident è non essere riusciti a ritornare tutte le tuple della tabella *CartsContainsProds*, si procederà alla modifica della *doRetrieveAllCartsContainsProds()* di *CartsContainsProdsDAO* che ha portato alla failure con conseguente riesecuzione del test case, di seguito il failure *testDoRetrieveAllCartsContainsProds()*:

```

Runs: 15/15      Errors: 0      Failures: 1
CartsContainsProdsDAOTest [Runner: JUnit 5] (0.378 s)
  testDoRetrieveByKey() (0.065 s)
  testDoRetrieveAllCartsContainsProds() (0.064 s)
  testDoRetrieveAllByCardIdCartsContainsProds() (0.061 s)
  testDoDeleteCartsContainsProds() (0.064 s)
  testDoSaveCartsContainsProds() (0.062 s)
  testDoUpdateCartsContainsProds() (0.059 s)
OrdersContainsProdsDAOTest [Runner: JUnit 5] (0.207 s)
Failure Trace
  org.opentest4j.AssertionFailedError: Le tuple di CartsContainsProdsDAO recuperate non sono identiche alle attese ==> array contents differ at index [1]
    at it.unisa.zyphysoprt.gestioneVendite.DAO.CartsContainsProdsDAOTest.testDoRetrieveAllCartsContainsProds(CartsContainsProdsDAOTest.java:88)
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
  
```

- *Failure testDoGetAddToCart():* l'incident è il non riuscire ad arrivare alla pagina attesa, si procederà alla modifica della response del metodo `doGet()` della servlet `AddToCartServlet` con conseguente riesecuzione del test case, di seguito il failure *testDoGetAddToCart():*

```

Runs: 1/1      Errors: 0      Failures: 1
Failure Trace
it.unisa.zyphysport.gestioneVendite.servlet.AddToCartServletTest [Runner: JUnit]
  testDoGetAddToCart (1,327 s)
    Argument(s) are different! Wanted:
      httpServletResponse.sendRedirect(
        "http://localhost/zyphyk-sport-web/cart.jsp"
      );
    => at it.unisa.zyphysport.gestioneVendite.servlet.AddToCartServletTest.testDoGetAddToCart(AddToCartServletTest.java:126)
    Actual invocations have different arguments:
      httpServletResponse.sendRedirect(
        "http://localhost/zyphyk-sport-web/index.jsp"
      );
    => at it.unisa.zyphysport.gestioneVendite.servlet.AddToCartServlet.doGet(AddToCartServlet.java:116)
    => at it.unisa.zyphysport.gestioneVendite.servlet.AddToCartServletTest.testDoGetAddToCart(AddToCartServletTest.java:126)
  
```

- *Failure testDoSaveProducts():* l'incident è il non salvataggio nella tabella della tupla, si procederà alla modifica del metodo `doSave()` di `ProductsDAO` con conseguente riesecuzione del test case, di seguito il failure *testDoSaveProducts():*

```

Runs: 14/14    Errors: 0      Failures: 1
Failure Trace
ManagesProdsDAOTest [Runner: JUnit 5] (1,541 s)
  SizesDAOTest [Runner: JUnit 5] (0,446 s)
  ProductsDAOTest [Runner: JUnit 5] (0,568 s)
    testDoRetrieveByKey0 (0,094 s)
    testDoUpdateProducts0 (0,098 s)
    testDoRetrieveAll0 (0,077 s)
    testDoDeleteProducts0 (0,100 s)
    testDoSaveProducts0 (0,111 s)
    testDoRetrieveAllExists0 (0,080 s)
    junit.framework.ComparisonFailure: value (table=PRODUCTS, row=5, col=sport) expected: <[tennis]> but was: <[calcetto]>
    => at org.dbunit.assertion.JUnitFailureFactory.createFailure(JUnitFailureFactory.java:39)
    => at org.dbunit.assertion.DefaultFailureHandler.createFailure(DefaultFailureHandler.java:105)
    => at org.dbunit.assertion.DefaultFailureHandler.handle(DefaultFailureHandler.java:240)
    => at org.dbunit.assertion.DbUnitAssertBase.failIfNecessary(DbUnitAssertBase.java:619)
    => at org.dbunit.assertion.DbUnitAssertBase.compareData(DbUnitAssertBase.java:602)
    => at org.dbunit.assertion.DbUnitAssertBase.compareData(DbUnitAssertBase.java:561)
    => at org.dbunit.assertion.DbUnitAssertBase.assertWithValueComparator(DbUnitAssertBase.java:439)
    => at org.dbunit.assertion.DbUnitAssert.assertEquals(DbUnitAssert.java:300)
    => at org.dbunit.assertion.DbUnitAssert.assertEquals(DbUnitAssert.java:268)
    => at org.dbunit.assertion.DbUnitAssert.assertEquals(DbUnitAssert.java:227)
    => at org.dbunit.Assertion.assertEquals(Assertion.java:140)
    => at it.unisa.zyphysport.gestioneCatalogo.DAO.ProductsDAOTest.testDoSaveProducts(ProductsDAOTest.java:109)
    => at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
    => at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
  
```

- *Failure testDoPostAddInCatalog():* l'incident è la non esecuzione della servlet dovuta da un parametro settato in modo errato, si procederà alla modifica del parametro con conseguente riesecuzione del test case, di seguito il failure *testDoPostAddInCatalog():*

```

Runs: 1/1      Errors: 0      Failures: 1
Failure Trace
it.unisa.zyphysport.gestioneCatalogo.servlet.AddInCatalogServletTest [Runner: JUnit]
  testDoPostAddInCatalog (1,296 s)
    Argument(s) are different! Wanted:
      httpServletResponse.sendRedirect(
        "http://localhost/zyphyk-sport-web/products.jsp?id=POP12"
      );
    => at it.unisa.zyphysport.gestioneCatalogo.servlet.AddInCatalogServletTest.testDoPostAddInCatalog(AddInCatalogServletTest.java:142)
    Actual invocations have different arguments:
      httpServletResponse.sendRedirect(
        "http://localhost/zyphyk-sport-web/products.jsp?id="
      );
    => at it.unisa.zyphysport.gestioneCatalogo.servlet.AddInCatalogServlet.doPost(AddInCatalogServlet.java:142)
    => at it.unisa.zyphysport.gestioneCatalogo.servlet.AddInCatalogServletTest.testDoPostAddInCatalog(AddInCatalogServletTest.java:142)
  
```

Integration testing

Nell'integration testing non sono stati riscontrati failure perché gli errori sono stati risolti in fase di unit testing.

System testing

Nel system testing abbiamo testato tutte le componenti e i sottosistemi tramite Selenium IDE. Anche in questo caso sono stati riscontrati alcuni failure nei test case specificati. I failure sono dovuti a errori nelle regEx e nelle funzioni javascript utilizzate per effettuare la validazione dei dati front-end.

Le failure riscontrate sono:

- *Failure TC_FR09-Registrazione_11*: l'incident è la corretta registrazione di un utente che non doveva andare a buon fine, poiché la lunghezza del cognome non è risultata corretta, si procederà con la modifica della regEx del campo di testo del cognome e con la riesecuzione del test, di seguito esecuzione failure:

Running 'TC_FR09-Registrazione_11'

1. open on http://localhost/zyphyk-sport-web/signUp.jsp OK
2. type on id=name with value Danilo OK
3. type on id=cognome with value Pi OK
4. type on id=username with value daniPiccolo OK
5. type on id=data with value 2001-06-12 OK
6. type on id=email with value danilopicc@gmail.com OK
7. type on id=password with value Ciaoitalia! OK
8. type on id=conferma with value Ciaoitalia! OK
9. click on id=submit OK
10. assertAlert on Cognome non valido Failed:
No response!!!!

'TC_FR09-Registrazione_11' ended with 1 error(s)

	Command	Target	Value
1	✓ open	http://localhost/zyphyk-sport-web/signUp.jsp	
2	✓ type	id=name	Danilo
3	✓ type	id=cognome	Pi
4	✓ type	id=username	daniPiccolo
5	✓ type	id=data	2001-06-12
6	✓ type	id=email	danilopicc@gmail.com
7	✓ type	id=password	Ciaoitalia!
8	✓ type	id=conferma	Ciaoitalia!
9	✓ click	id=submit	
10	✗ assert alert	Cognome non valido	

- *Failure TC_FR08-Acquisto_12*: l'incident è la corretta esecuzione della procedura di acquisto che in questo caso non doveva essere eseguita, in quanto il formato della carta non è corretto, si procederà con la modifica della regex del campo di testo del numero di carta con la conseguente riesecuzione del caso di test, di seguito esecuzione failure:

	Command	Target	Value
1	✓ open	http://localhost/zyphyk-sport-web/checkOut.jsp	
2	✓ type	id=via	Via Roma
3	✓ type	id=numCivico	54
4	✓ type	id=città	Salerno
5	✓ type	id=provincia	SA
6	✓ type	id=cc-name	Ciro Vitale
7	✓ type	id=cc-number	53332837213378AQ
8	✓ type	id=cc-cvv	321
9	✓ type	id=cc-expiration	2027-08
10	✓ click	id=submit	
11	✗ assert alert	Numero carta non valido	

Running 'TC_FR08-Acquisto_12'

1. open on http://localhost/zyphyk-sport-web/checkOut.jsp OK
2. type on id=via with value Via Roma OK
3. type on id=numCivico with value 54 OK
4. type on id=città with value Salerno OK
5. type on id=provincia with value SA OK
6. type on id=cc-name with value Ciro Vitale OK
7. type on id=cc-number with value 53332837213378AQ OK
8. type on id=cc-cvv with value 321 OK
9. type on id=cc-expiration with value 2027-08 OK
10. click on id=submit OK
11. assertAlert on Numero carta non valido Failed:
No response!!!!

'TC_FR08-Acquisto_12' ended with 1 error(s)

- *Failure TC_FR01-InserimentoProdottoInCatalogo_06*: l'incident è il corretto inserimento di un prodotto nel catalogo che non deve essere inserito poiché non è stata selezionata nessuna taglia, si procederà con la riesecuzione del test case dopo aver modificato la funzione javascript che effettua il controllo effettuato sulla checkbox delle taglie, di seguito esecuzione failure:

Running 'TC_FR01-InserimentoProdottoInCatalogo_06'

1. open on `http://localhost/zyphyk-sport-web/addInCatalog.jsp` OK
2. type on `id=productId` with value `FW290` OK
3. type on `id=nomeProd` with value `Nike Air Zoom` OK
4. type on `id=price` with value `95` OK
5. type on `id=inputImage` with value `C:\fakepath\example.jpg` OK
6. click on `id=submit` OK
7. assertAlert on `Selezionare almeno una taglia da inserire` **Failed:**
No response!!!!

'TC_FR01-InserimentoProdottoInCatalogo_06' ended with 1 error(s)

	Command	Target	Value
1	✓ open	<code>http://localhost/zyphyk-sport-web/addInCatalog.jsp</code>	
2	✓ type	<code>id=productId</code>	<code>FW290</code>
3	✓ type	<code>id=nomeProd</code>	<code>Nike Air Zoom</code>
4	✓ type	<code>id=price</code>	<code>95</code>
5	✓ type	<code>id=inputImage</code>	<code>C:\fakepath\example.jpg</code>
6	✓ click	<code>id=submit</code>	
7	✗ assert alert	<code>Selezionare almeno una taglia da inserire</code>	

Test Summary Report

Il testing ci ha permesso di scovare un sacco di failure nel sistema che altrimenti, senza l'aiuto dei tool, sarebbero stati più difficili da trovare. Gli incident sono stati tutti risolti con successo e dopo una riesecuzione di tutti i casi di test, non sono state riscontrate failure come riportato nel TRR (Test Regression Report).

Per quanto riguarda le tecniche di testing utilizzate, per lo Unit Testing è stato utilizzato Weak Equivalence Class Testing che ci ha permesso di avere delle classi di test dove è bastato testare un parametro differente da quello atteso per verificare se il test ha avuto successo o meno, in questo caso possiamo dire che è stato efficace ma ovviamente potremmo aver mancato alcuni casi di test che avrebbero potuto portare a failure.

Per l'integration testing abbiamo utilizzato la strategia bottom-up che ci ha consentito di testare prima le componenti di basso livello per arrivare poi a testare tutto il sistema. Questa strategia ci ha portato ad avere meno problemi nelle fasi successive allo unit testing poiché i DAO e le Servlet risultavano essere già testate e funzionanti e ad anticipare dunque problemi di testing che si potevano riscontrare in seguito.

Infine per il System testing abbiamo utilizzato Category Partition, che ci ha consentito di raggruppare in modo specifico le classi di test in categorie che siano più efficienti e comprensibili.