



TSR (Test Summary Report)

Smart Cooking

Riferimento	
Versione	0.2
Data	08/11/2021
Destinatario	Prof.ssa F. Ferrucci
Presentato da	Benedetta del Giudice, Daniele Colucci, Marco Luisi, Antonio Alfano, Anton Zolotko
Approvato da	



Revision History

Data	Versione	Descrizione	Autori
01/11/2021	0.1	Prima stesura	Marco Luisi
03/11/2021	0.2	Revisione	Benedetta del Giudice



Sommario

Revision History	2
1.Introduzione	4
1.1 Identificativo del documento	4
1.2 Scopo	4
1.3 Riferimenti	4
2. Risultati di JUnit per il Model	4
2.1 Features testate	5
2.1.1 Panoramica dei risultati del test delle classi	5
3. Risultati Mockito per il Controller	6
3.1 Features testate	6
3.1.1 Panoramica dei risultati del test delle classi	6
4. Risultati Selenium per il View	8
4.1 Features Testate	8
4.1.1 Panoramica dei risultati del test delle pagine jsp	8
4.2 Features non Testate	10
5. COVERAGE:	10
6. Riepilogo del testing	11
7. Glossario	11



1.Introduzione

Questo documento ha lo scopo di riportare i risultati dell'esecuzione dei test case di unità delle varie componenti del software web SmartCooking. In particolare, sono state testate le classi del package model e controller con l'utilizzo di JUnit., e successivamente è stato effettuato il testing di sistema con Selenium.

1.1 Identificativo del documento

Questo documento è caratterizzato da alcune proprietà che lo definiscono univocamente; tali proprietà sono: data ed autore. Il documento contiene il nome di chi si è occupato della stesura dello stesso.

1.2 Scopo

Lo scopo di questo documento è quello di fornire una presentazione dei casi di test di unità per il software SmartCooking. I vari membri del team si sono impegnati nel verificare che le singole unità (classi e metodi) abbiano il comportamento atteso.

1.3 Riferimenti

- NV02_ODD_V0.2
- NC02_RAD_V2.2
- NC02_SDD_V0.4
- NC02_TPD_V0.2
- NC02_TCS_V0.2
- B.Bruegge, A.H. Dutoit, Object Oriented Software Engineering – Using UML, Patterns and Java, Prentice Hall.
- Slides del corso, presenti sulla piattaforma e-learning.

2. Risultati di JUnit per il Model

Di seguito sono riportati i risultati dei test di unità per le classi del Package model



2.1 Features testate

Il testing di unità si propone di effettuare il test delle classi presenti all'interno del Package model.

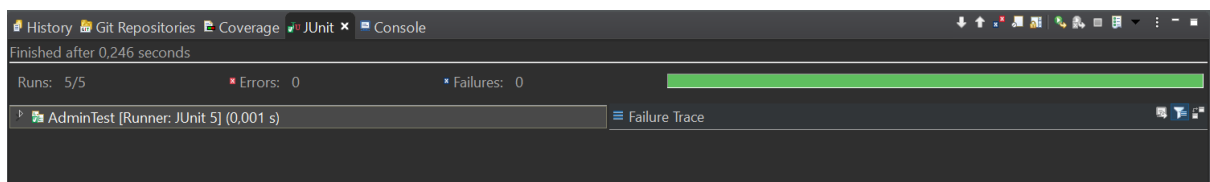
In particolare, saranno eseguiti:

- Admin
- Utente
- Video

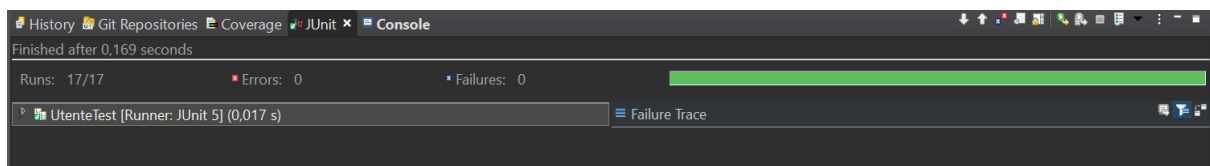
2.1.1 Panoramica dei risultati del test delle classi

In nessuna classe sono stati riscontrati errori e/o failure. Di seguito sono riportati i risultati dell'esecuzione di ogni classe.

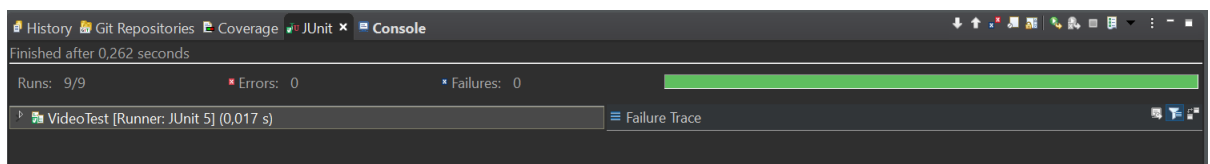
Classe Admin



Classe Utente



Classe Video





3. Risultati Mockito per il Controller

Di seguito sono riportati i risultati dei test di unità per le classi del Package controller.

3.1 Features testate

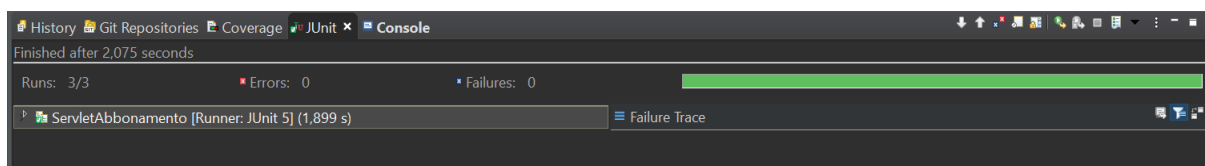
Il testing di unità si propone di effettuare il test delle classi presenti all'interno del Package controller.

In particolare, saranno eseguiti:

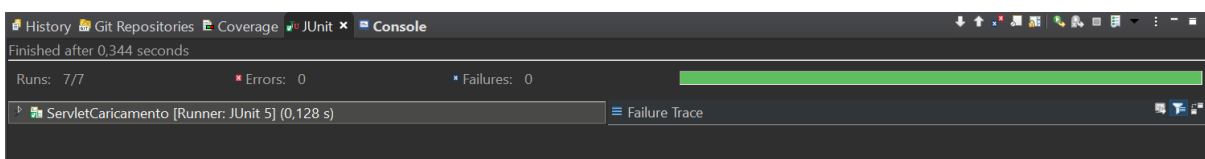
- Abbonamento
- CaricamentoVideo
- Login
- ModificaDatiUtente
- RemoveVideoAdmin
- UpdateVideoAdmin
- Registrazione
- VideoCategoria
- VideoKeyword

3.1.1 Panoramica dei risultati del test delle classi

Classe Abbonamento

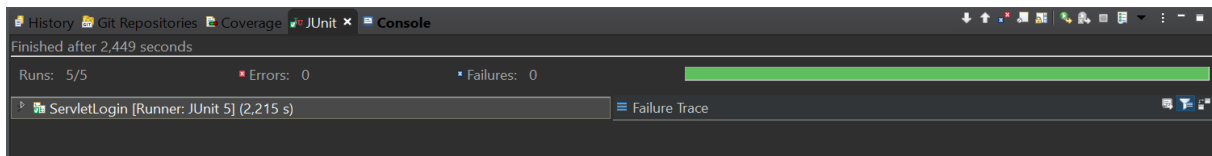


Classe CaricamentoVideo

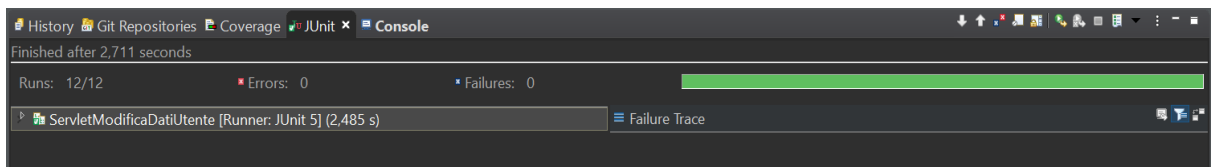




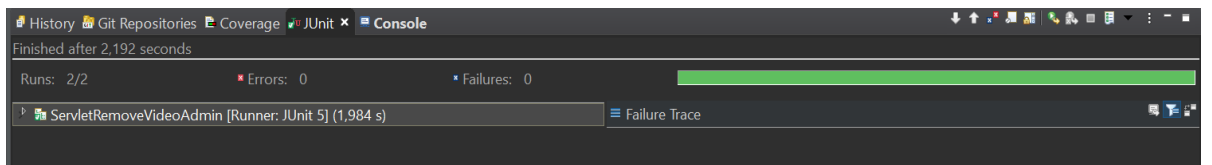
Classe Login



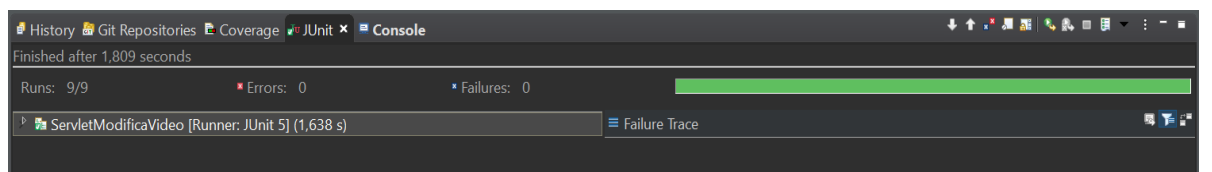
Classe ModificaDatiUtente



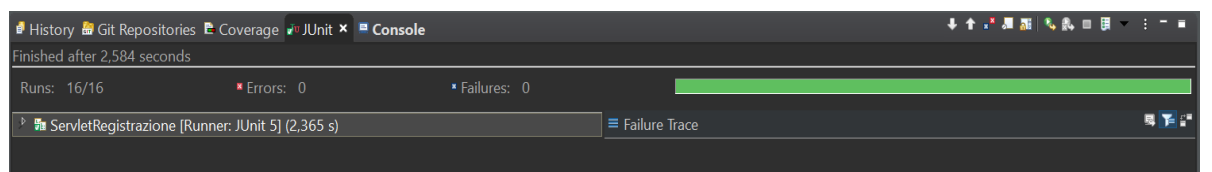
Classe RemoveVideoAdmin



Classe UpdateVideoAdmin

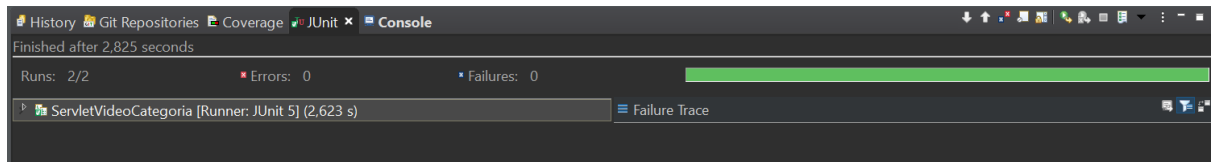


Classe Registrazione

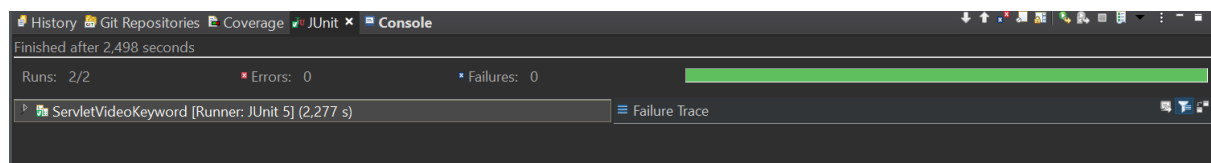




Classe VideoCategoria



Classe VideoKeyword



4. Risultati Selenium per il View

Di seguito sono riportati i risultati dei test di sistema per le pagine del Package view.

4.1 Features Testate

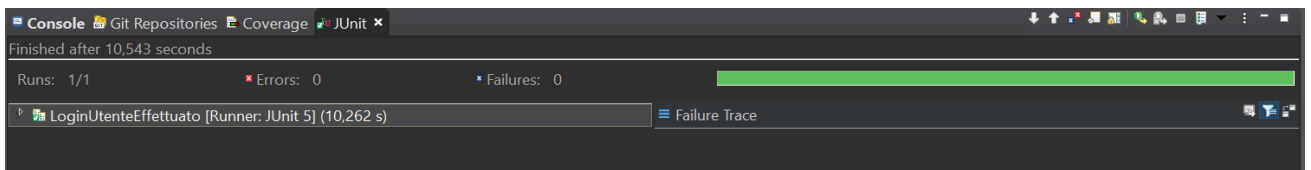
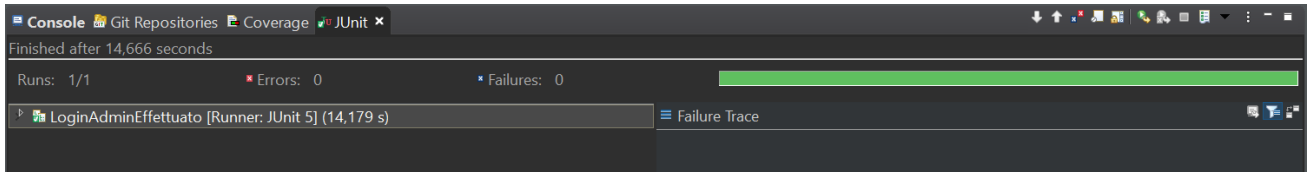
Il testing di sistema si propone di eseguire i test case pianificati in precedenza. In particolare, saranno eseguiti:

- TC_1 Registrazione
- TC_2 Login
- TC_4 Abbonamento
- TC_5 CaricamentoVideo
- TC_7 EliminaVideo

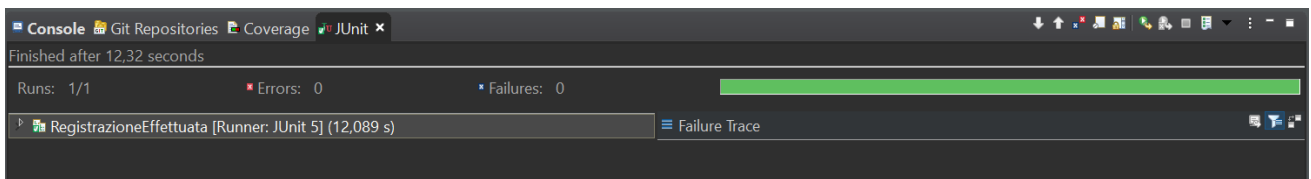
4.1.1 Panoramica dei risultati del test delle pagine jsp



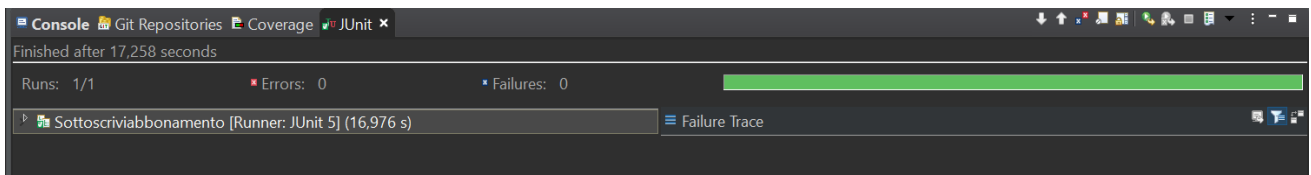
login.jsp



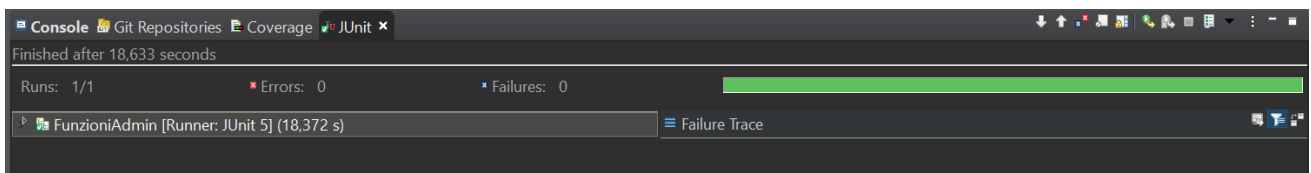
registrazione.jsp



abbonamento.jsp



admin.jsp





4.2 Features non Testate

Alcuni dei test precedentemente pianificati non sono stati testati perché l'implementazione non ha permesso un testing esaustivo. Di seguito sono riportati i test case che non sono stati eseguiti:

- TC_3 ModificaDatiUtente
- TC_6 ModificaVideo
- TC_8 Ricerca video per categoria
- TC_9 Ricerca video per keyword

5. COVERAGE:

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
provaSmartCooking3	87,7 %	5.640	794	6.434
src	87,7 %	5.640	794	6.434
systemTesting	100,0 %	836	0	836
Model	91,1 %	267	26	293
Admin.java	100,0 %	31	0	31
AdminRepositoryImpl.java	100,0 %	12	0	12
Utente.java	100,0 %	86	0	86
Video.java	91,0 %	61	6	67
UtenteRepositoryImpl.java	86,8 %	33	5	38
VideoRepositoryImpl.java	74,6 %	44	15	59
Test	87,5 %	2.565	368	2.933
Control	86,2 %	1.040	167	1.207
Login.java	99,4 %	168	1	169
ModificaDatiUtente.java	99,3 %	141	1	142
VideoCategoria.java	98,4 %	61	1	62
Abbonamento.java	97,9 %	46	1	47
RemoveVideoAdmin.java	97,6 %	40	1	41
VisualizzazioneListaUtenti.java	96,9 %	31	1	32
VisualizzazioneListaVideo.java	96,9 %	31	1	32
Logout.java	93,8 %	15	1	16
Registrazione.java	93,6 %	161	11	172
UpdateVideoAdmin.java	93,3 %	168	12	180
VideoKeyword.java	90,2 %	46	5	51
CaricamentoVideo.java	58,9 %	132	92	224
ModifyVideoAdmin.java	0,0 %	0	39	39
DAO	80,0 %	932	233	1.165



6. Riepilogo del testing

	Numero di componenti testate	Numero di errori trovati	Numero di errori corretti	Numero di componenti non testate
model	3	0	0	1
view	4	0	0	2
control	9	0	0	2

7. Glossario

- RAD (Requirement Analysis Document): Documento di Raccolta e analisi dei Requisiti; contiene l'elenco dei requisiti funzionali e non funzionali individuati in fase di individuazione degli stessi e la loro analisi sotto forma di scenari e casi d'uso. I mock-up mostrano una possibile implementazione dell'interfaccia del sistema.
- SDD (System Design Document): Documento all'interno del quale viene riportata la progettazione del sistema come risultato di una prima fase di modellazione: contiene una suddivisione ad alto livello del sistema nei sottosistemi che lo comporranno.
- ODD (Object Design Document): Documento che riporta e analizza gli oggetti che compongono il sistema analizzando le componenti a più basso livello, riportandole così come saranno implementate.
- TCD (Test Case Document): Documento che riporta i casi di test pianificati.