

Test Plan

Versione	Data	Descrizione	Autori
0.1	12/12/2019	Prima stesura	Michelantonio Panichella
0.2	9/1/2020	Aggiornamento sottosistemi	Gianluca Pirone
0.3	16/1/2020	Riformulazione test cases	Umberto Loria
0.4	19/1/2020	Esecuzione test	Team

Indice

1. [Introduzione](#)
2. [Relazioni con gli altri documenti](#)
 - i. [Relazione con RAD](#)
 - ii. [Relazione con SDD](#)
 - iii. [Relazione con ODD](#)
3. [Panoramica del sistema](#)
4. [Funzionalità da testare](#)
5. [Criteri Pass/Field](#)
6. [Approccio](#)
7. [Sospensione e ripresa](#)
 - i. [Sospensione](#)
 - ii. [Ripresa](#)
8. [Materiale per il testing](#)
9. [Casi di test](#)
 - i. [Ricerca](#)
 - a. [RicercaFilm](#)
 - b. [RicercaArtista](#)
 - c. [RicercaUtente](#)
 - ii. [Account](#)
 - a. [CreareAccount](#)
 - b. [AutenticareAccount](#)
 - c. [CambiarePassword](#)
 - iii. [Amicizia](#)
 - a. [InviareRichiestaAmicizia](#)
 - b. [AccettareRichiestaAmicizia](#)
 - c. [RifiutareRichiestaAmicizia](#)
 - iv. [Film](#)

- a. [AggiungereGiudizio](#)
- b. [ModificareGiudizio](#)
- c. [RimuovereGiudizio](#)
- v. [Gestione](#)
 - a. [AggiungereFilm](#)
 - b. [AggiungereArtista](#)
 - c. [AggiungereGenere](#)
 - d. [ModificareGenere](#)
 - e. [RimuovereGenere](#)
 - f. [AggiornareGeneriFilm](#)

Introduzione

Molte volte, anche credendo di aver progettato un software nel migliore dei modi, ci troviamo di fronte a situazioni nelle quali basta un piccolo evento, una piccola azione fatta da un attore, per portare un malfunzionamento a tutto il sistema. Ma come facciamo a trattare questi errori?

Esistono principalmente tre diversi modi per affrontare il problema:

- Prevenzione di errori (prima che rilasciamo un sistema)
- Rilevazione di errori (quando un sistema è in esecuzione)
- Recupero di errori (dopo che un sistema è stato rilasciato)

Tra tutte queste tecniche, quella utilizzata per costruire nel migliore dei modi Moovie, noi abbiamo scelto quella di "Rilevazione di errori" in cui è presente l'attività di Testing.

In questo documento infatti, andremo ad illustrare come attraverso questa attività di testing, cercheremo di rilevare tutti gli errori che nel sistema sono presenti.

Il testing sarà eseguito sui seguenti sottosistemi:

- Ricerca
- Account
- Amicizia
- Film
- Gestione

Relazioni con gli altri documenti

Il documento di "Test Plan" è strettamente collegato a:

- **Test Case Specification:** documento in cui tutti i casi di test presentati nel test plan, verranno specificati nel dettaglio, concentrandoci non solo sull'oracolo, ma anche sul flusso di eventi di ogni caso di test;
- **Test Execution Report:** documento in cui vengono illustrate le esecuzioni di tutti i casi di test presenti nel test plan;
- **Test Log:** documento in cui vengono riportati tutti i risultati derivanti dalle esecuzioni dei test case.

Relazione con RAD

Il Test Plan è strettamente collegato con il documento di Analisi dei requisiti in quanto dai requisiti funzionali emersi durante l'attività di analisi abbiamo individuato le principali funzionalità che il sistema deve avere. Inoltre, dato che abbiamo deciso di utilizzare la tecnica di "Category Partition" per la scelta dei casi di test, le categorie, i parametri e scelte prese sono strettamente correlate agli Use Cases Presenti nel RAD.

Relazione con SDD

Il Test Plan è strettamente collegato con il documento di "System Design" poiché dovrà rispettare tutte le funzionalità ottenute dai vari sottosistemi, facendo riferimento anche alla suddivisione fatta: Presentation Layer, Application Layer e Data Layer.

Relazione con ODD

Il Test Plan è strettamente collegato con il documento di "Object Design" poiché dovrà essere quanto più conforme possibile alle interfacce definite per ogni classe.

Panoramica del sistema

Per una migliore progettazione della nostra Web application, il sistema è stato suddiviso in vari Layer che sono:

Presentation Layer

Application Layer

Data Layer

Questa suddivisione è stata fatta per ottenere basso accoppiamento (in modo tale che ogni qual volta si modifichi una componente del sistema non c'è la necessità di modificare tutte le altre accoppiate ad essa) ed alta coesione (in modo tale che le classi nel sottosistema svolgono compiti simili e sono correlate tra loro).

Funzionalità da testare

In base alla suddivisione in sottosistemi che si è fatta, per ogni sottosistema abbiamo scelto di testare varie funzionalità, qui di seguito quelle individuate:

- Per il sottosistema Ricerca
 - RicercaFilm
 - RicercaArtista
 - RicercaUtente
- Per il sottosistema Account
 - CreareAccount
 - AutenticareAccount
 - CambiarePassword
- Per il sottosistema Amicizia
 - InviareRichiestaAmicizia
 - AccettareRichiestaAmicizia

- RifiutareRichiestaAmicizia
- Per il sottosistema Film
 - AggiungereGiudizio
 - ModificareGiudizio
 - RimuovereGiudizio
- Per il sottosistema Gestione
 - AggiungereFilm
 - AggiungereArtista
 - AggiungereGenere
 - ModificareGenere
 - RimuovereGenere
 - AggiornareGeneriFilm

Criteri Pass/ Field

Durante l'attività di testing, cercheremo di rilevare errori nel modo più pianificato possibile. Lo scopo di questa attività non è quello di dimostrare che nel sistema non ci sono "failure" ma quello di trovarne la presenza andando ad esercitare ogni funzionalità che il cliente si aspetta.

Affinché ogni funzionalità "passi" correttamente il test, è necessario che l'output derivante da esso sia conforme alle specifiche (ORACOLO) descritte in precedenza dallo sviluppatore. Nel caso non fosse così, la componente non passerà il test e si dovrà procedere per una correzione.

Approccio

Tra tutte le tecniche di verifica di un software, una che ricopre molta importanza è quella di "Analisi dinamica". Secondo questa tecnica, esistono due modi per interfacciarsi verso gli errori del sistema: "Black-box testing" e "White-box testing". Il testing che si è scelto di eseguire è quello chiamato "black box" secondo il quale la selezione dei test è fatta sulle specifiche delle funzionalità e non sull'effettiva implementazione. Per procedere all'attività di testing si è deciso di applicare una strategia "Bottom up" secondo la quale andremo a testare dapprima le componenti più piccole ed elementari che compongono il nostro programma, poi andremo ad integrare le stesse e vedere il loro corretto funzionamento e infine procedere al test di sistema. Le varie componenti del testing che sono state individuate sono le seguenti:

- Test di unità
 - Testare i vari sottosistemi individualmente, testare singolarmente le varie funzionalità dei sottosistemi. Durante questa fase verranno testati singolarmente DAO e controller.
- Test di integrazione
 - Unire alcuni sottosistemi e testarli con tutte le loro funzionalità integrate. Durante questa fase, cercheremo di rilevare gli errori derivanti dalla composizione di alcuni sottosistemi, nel nostro caso verrà collaudata l'effettiva funzionalità derivante dall'integrazione dei Controller.
- Test di sistema
 - Testare l'intero funzionamento del sistema. Durante questa fase cercheremo di rilevare gli errori derivanti dall'intero sistema e dall'integrazione, dunque, di tutti i sottosistemi prima sviluppati.

Sospensione

Ci sarà una sospensione dell'attività di testing solo nel momento in cui avremo testato ogni funzionalità prevista rispettando nel modo quanto più corretto possibile quelli che sono i tempi e i costi da noi previsti.

Ripresa

Ogni qual volta si individueranno dei bug nelle funzionalità sarà necessaria una correzione di essi. Per verificare che la correzione non abbia portato ulteriori danni al sistema verrà ripresa l'attività di testing, riproponendo gli stessi casi di test che ci hanno condotto al problema.

Materiale per il testing

Per eseguire i vari casi di test faremo uso di:

- un database che conterrà tutti i dati;
- un client che si occuperà di interfacciarsi con le funzionalità da testare;
- un server che dovrà eseguire tutte le richieste del client.

Casi di test

Ricerca

RicercaFilm

PARAMETRO:	RicercaFilm
Lunghezza[LRicercaFilm]:	1. < 1[Errore] 2. > 0[Successo_LRicercaFilm]

Test cases

Codice	Combinazione	Esito
TC_1.1_1	LRicercaFilm_1	Errore
TC_1.1_2	LRicercaFilm_2	Corretto

RicercaArtista

PARAMETRO:	RicercaArtista
Lunghezza[LRicercaArtista]:	1. < 1[Errore] 2. > 0[Successo_LRicercaArtista]

Test cases

Codice	Combinazione	Esito
TC_1.2_1	LRicercaArtista_1	Errore
TC_1.2_2	LRicercaArtista_2	Corretto

RicercaUtente

PARAMETRO:	Utente
Loggato[LoUtente]:	1. L'utente che vuole effettuare la ricerca non è loggato[Errore] 2. L'utente che vuole effettuare la ricerca è loggato[Successo_LoUtente]

PARAMETRO:	RicercaUtente
Lunghezza[LRicercaUtente]:	1. < 1[Errore] 2. > 0[Successo_LRicercaUtente]

Test cases

Codice	Combinazione	Esito
TC_1.3_1	LoUtente_1	Errore
TC_1.3_2	LoUtente_2, LRicercaUtente_1	Errore
TC_1.3_3	LoUtente_2, LRicercaUtente_2	Corretto

Account

CreareAccount

PARAMETRO:	Nome
FORMATO:	^[a-zA-Z' àèèìòù]+\$
Lunghezza[LNome]:	1. <2 or >50 [Errore] 2. >=2 and <=50 [Successo_LNome]
Formato[FNome]:	1. Non rispecchia il formato[if_LNome_OK][Errore] 2. Rispetta il formato[if_FNome_OK][Successo_Nome]

PARAMETRO:	Cognome
FORMATO:	^[a-zA-Z' àèèìòù]+\$
Lunghezza[LCognome]:	1. <2 or >50 [Errore] 2. >=2 and <=50 [Successo_LCognome]
Formato[FNome]:	>1. Non rispecchia il formato [if_LCognome_OK][Errore] 2. Rispetta il formato[if_FCognome_OK][Successo_Cognome]

PARAMETRO:	Email
FORMATO:	^[\\w\\.-]+@[\\w\\.-]+\\.\\w{2,4}\$
Lunghezza[LEmail]:	1. <5 or >50 [Errore] 2. >=5 & <=50 [Successo_LEmail]

PARAMETRO:	Email
Formato[FEmail]:	1. Non rispecchia il formato [if_LEmail_OK][Errore] 2. Rispetta il formato[if_FEmail_OK][Successo_FEmail]
Presenza[PEmail]:	1. Email già in uso [if_FEmail_OK][Errore] 2. Email non in uso [if_FEmail_OK][Successo_Email]

PARAMETRO:	Password
Lunghezza[LPASSWORD]:	1. <6 or >16 [Errore] 2. >=6 & <=16 [Successo_LPASSWORD]

PARAMETRO:	Copassword
Lunghezza[LCopassword]:	1. <6 or >16 [Errore] 2. >=6 & <=16 [Successo_LCopassword]
Corrispondenza[CPASSWORD]:	1. La password inserita è diversa da quella scelta[if_LCopassword_ok] [Errore] 2. La password inserita corrisponde a quella scelta[if_CPASSWORD_OK] [Successo_Password]

Test cases

Codice	Combinazione	Esito
TC_2.1_1	LNome_1	Errore
TC_2.1_2	LNome_2, FNome_1	Errore
TC_2.1_3	LNome_2,FNome_2 LCognome_1	Errore
TC_2.1_4	LNome_2,FNome_2 LCognome_2, FCognome_1	Errore
TC_2.1_5	LNome_2,FNome_2 LCognome_2, FCognome_2 LEmail_1	Errore
TC_2.1_6	LNome_2,FNome_2 LCognome_2, FCognome_2 LEmail_2, FEmail_1	Errore
TC_2.1_7	LNome_2,FNome_2 LCognome_2, FCognome_2 LEmail_2, FEmail_2, PEmail_1	Errore
TC_2.1_8	LNome_2,FNome_2 LCognome_2, FCognome_2 LEmail_2, FEmail_2, PEmail_2, LPASSWORD_1	Errore

Codice	Combinazione	Esito
TC_2.1_9	LNome_2,FNome_2 LCognome_2, FCognome_2 LEmail_2, FEmail_2, PEmail_2, LPassword_2, LCopassword_1	Errore
TC_2.1_10	LNome_2,FNome_2 LCognome_2, FCognome_2 LEmail_2, FEmail_2, PEmail_2, LPassword_2, LCopassword_2	Corretto

AutenticareAccount

PARAMETRO:	Email
FORMATO:	^[\\w\\.-]+@[\\w\\.-]+\\.\\w{2,4}\$
Lunghezza[LEmail]:	1. <5 or >50 [Errore] 2. >=5 & <=50 [Successo_LEmail]
Formato[FEmail]:	1. Non rispetta il formato [if_LEmail_OK][Errore] 2. Rispetta il formato[if_FEmail_OK][Successo_Cognome]
Corrispondenza[CEmail]:	1. Email inserita non corrisponde[if_FEmail_OK][Errore] 2. Email inserita corrisponde[if_FEmail_OK][Successo_Email]

PARAMETRO:	Password
Lunghezza[LPassword]:	1. <6 or >16 [Errore] 2. >=6 & <=16 [Successo_LPassword]
Corrispondenza[CPassword]:	1. La password inserita non corrisponde[if_LCopassword_ok][Errore] 2. La password inserita corrisponde a quella scelta[if_CPassword_OK] [Successo_Password]

Test cases

Codice	Combinazione	Esito
TC_2.2_1	LEmail_1	Errore
TC_2.2_2	LEmail_2, FEmail_1	Errore
TC_2.2_3	LEmail_2, FEmail_2 CEmail_1	Errore
TC_2.2_4	LEmail_2, FEmail_2 CEmail_2, LPassword_1	Errore
TC_2.2_5	LEmail_2, FEmail_2 CEmail_2, LPassword_2, CPassword_1	Errore
TC_2.2_6	LEmail_2, FEmail_2 CEmail_2, LPassword_2, CPassword_2	Corretto

CambiarePassword

PARAMETRO:	Utente
Loggato[LoUtente]:	1. L'utente che vuole cambiare la password non è loggato[Errore] 2. L'utente che vuole cambiare la password è loggato[Successo_LoUtente]

PARAMETRO:	PasswordVecchia
Lunghezza[LPASSWORDVecchia]:	1. < 6 or > 16[Errore] 2. >=6 & <=16[Successo_LPASSWORDVecchia]
Corrispondenza[CPASSWORDVecchia]:	1. La password è diversa da quella precedente[if_LPASSWORDVecchia ok][Errore] 2. La password inserita corrisponde a quella precedente[if_CPASSWORDVecchia OK] [Successo_CPASSWORDVecchia]

PARAMETRO:	PasswordNuova
Lunghezza[LPASSWORDNuova]:	1. < 6 or > 16[Errore] 2. >=6 & <=16[Successo_LPASSWORDNuova]

Test cases

Codice	Combinazione	Esito
TC_2.3_1	LoUtente_1	Errore
TC_2.3_2	LoUtente_2, LPASSWORDVecchia_1	Errore
TC_2.3_3	LoUtente_2, LPASSWORDVecchia_2, CPASSWORDVecchia_1	Errore
TC_2.3_4	LoUtente_2, LPASSWORDVecchia_2, CPASSWORDVecchia2, LPASSWORDNuova_1	Errore
TC_2.3_5	LoUtente_2, LPASSWORDVecchia_2, CPASSWORDVecchia2, LPASSWORDNuova_2	Corretto

Amicizia

InviareRichiestaAmicizia

PARAMETRO:	UtenteFrom
Loggato[LoUtenteFrom]:	1. L'utente che invia l'amicizia non è loggato[Errore] 2. L'utente che invia l'amicizia è loggato [Successo_LoUtenteFrom]

PARAMETRO:	UtenteTo
Esistenza[EUtenteTo]:	1. L'utente a cui inviare l'amicizia non esiste[Errore] 2. L'utente a cui inviare l'amicizia esiste[Successo_EUtenteTo]

Test cases

Codice	Combinazione	Esito
TC_3.1_1	LoUtenteFrom_1	Errore

Codice	Combinazione	Esito
TC_3.1_2	LoUtenteFrom_2, EUtenteTo_1	Errore
TC_3.1_3	LoUtenteFrom_2, EUtenteTo_2	Corretto

AccettareRichiestaAmicizia

PARAMETRO:	UtenteTo
Loggato[LoUtenteTo]:	1. L'utente che accetta l'amicizia non è loggato[Errore] 2. L'utente che accetta l'amicizia è loggato [Successo_LoUtenteTo]

PARAMETRO:	UtenteFrom
Esistenza[EUtenteFrom]:	1. L'utente che invia l'amicizia non esiste[Errore] 2. L'utente che invia l'amicizia esiste[Successo_EUtenteFrom]

Test cases

Codice	Combinazione	Esito
TC_3.2_1	LoUtenteTo_1	Errore
TC_3.2_2	LoUtenteTo_2, EUtenteFrom_1	Errore
TC_3.2_3	LoUtenteTo_2, EUtenteFrom_2	Corretto

RifiutareRichiestaAmicizia

PARAMETRO:	UtenteTo
Loggato[LoUtenteTo]:	1. L'utente che rifiuta l'amicizia non è loggato[Errore] 2. L'utente che rifiuta l'amicizia è loggato [Successo_LoUtenteTo]

PARAMETRO:	UtenteFrom
Esistenza[EUtenteFrom]:	1. L'utente che invia l'amicizia non esiste[Errore] 2. L'utente che invia l'amicizia esiste[Successo_EUtenteFrom]

Test cases

Codice	Combinazione	Esito
TC_3.3_1	LoUtenteTo_1	Errore
TC_3.3_2	LoUtenteTo_2, EUtenteFrom_1	Errore
TC_3.3_3	LoUtenteTo_2, EUtenteFrom_2	Corretto

Film

AggiungereGiudizio

PARAMETRO:	Utente
------------	--------

PARAMETRO: Loggato[LoUtente]:	1. L'utente che vuole aggiungere un giudizio non è loggato[Errore]
	2. L'utente che vuole aggiungere un giudizio è loggato[Successo_LoUtente]

PARAMETRO:	Voto
Selezione[SVoto]:	1. L'utente ha selezionato un valore <1 or >10 [Errore] 2. L'utente ha selezionato un valore >=1 & <=10 [Successo_SVoto]

PARAMETRO:	Film
Presenza[PFilm]:	1. Il film non è presente[Errore] 2. Il film è presente[Successo_PFilm]

Test cases

Codice	Combinazione	Esito
TC_4.1_1	LoUtente_1	Errore
TC_4.1_2	LoUtente_2, SVoto_1	Errore
TC_4.1_3	LoUtente_2, SVoto_2, PFilm_1	Errore
TC_4.1_4	LoUtente_2, SVoto_2, PFilm_2	Corretto

ModificareGiudizio

PARAMETRO:	Utente
Loggato[LoUtente]:	1. L'utente che vuole modificare un giudizio non è loggato[Errore] 2. L'utente che vuole modificare un giudizio è loggato[Successo_LoUtente]

PARAMETRO:	Voto
Selezione[SVoto]:	1. L'utente ha selezionato un valore <1 or >10 2. L'utente ha selezionato un valore >=1 & <=10 e diverso da quello precedentemente assegnato[Successo_SVoto]

PARAMETRO:	Film
Presenza[PFilm]:	1. Il film non è presente[Errore] 2. Il film è presente[Successo_PFilm]

Test cases

Codice	Combinazione	Esito
TC_4.2_1	LoUtente_1	Errore
TC_4.2_2	LoUtente_2, SVoto_1	Errore
TC_4.2_3	LoUtente_2, SVoto_2, PFilm_1	Errore
TC_4.2_4	LoUtente_2, SVoto_2, PFilm_2	Corretto

RimuovereGiudizio

PARAMETRO:	Utente
Loggato[LoUtente]:	1. L'utente che vuole rimuovere un giudizio non è loggato[Errore] 2. L'utente che vuole rimuovere un giudizio è loggato[Successo_LoUtente]

PARAMETRO:	Film
Presenza[PFilm]:	1. Il film non è presente[Errore] 2. Il film è presente[Successo_PFilm]

PARAMETRO:	Voto
Presenza[PVoto]:	1. L'utente non ha assegnato nessun giudizio in precedenza 2. L'utente ha assegnato un giudizio in precedenza[Successo_PVoto]

Test cases

Codice	Combinazione	Esito
TC_4.3_1	LoUtente_1	Errore
TC_4.3_2	LoUtente_2, PFilm_1	Errore
TC_4.3_3	LoUtente_2, PFilm_2, SVoto_1	Errore
TC_4.3_4	LoUtente_2, PFilm_2, SVoto_2	Corretto

Gestione

AggiungereFilm

PARAMETRO:	Utente
Loggato[LoUtente]:	1. L'utente che vuole aggiungere un film non è loggato[Errore] 2. L'utente che vuole aggiungere un film è loggato[Successo_LoUtente]
Gestore[GeUtente]:	1. L'utente che vuole aggiungere un film non è gestore[if_LoUtente_OK][Errore] 2. L'utente che vuole aggiungere un film è gestore[if_GeUtente_OK][Successo_Utente]

PARAMETRO:	Titolo
Lunghezza[LTitolo]:	1. < 3 or > 100[Errore] 2. > 2 & < 101[Successo_LTitolo]

PARAMETRO:	Durata
FORMATO:	^\d+\$
Formato[FDurata]:	1. Non rispetta il formato[Errore] 2. Rispetta il formato[Successo_FDurata]

PARAMETRO:	Anno
FORMATO:	^[0-9]{4}\$
Formato[FAnno]:	1. Non rispetta il formato[Errore] 2. Rispetta il formato[Successo_FAnno]

PARAMETRO:	Descrizione
Lunghezza[LDescrizione]:	1. < 10[Errore] 2. > 9[Successo_LDescrizione]

PARAMETRO:	Copertina
FORMATO:	JPG or PNG
Caricamento[CaCopertina]:	1. Copertina non caricata[Errore] 2. Copertina caricata[Successo_CaCopertina]

Test cases

Codice	Combinazione	Esito
TC_5.1_1	LoUtente_1	Errore
TC_5.1_2	LoUtente_2, GeUtente_1	Errore
TC_5.1_3	LoUtente_2, GeUtente_2, LTitolo_1	Errore
TC_5.1_4	LoUtente_2, GeUtente_2, LTitolo_2, FDurata_1	Errore
TC_5.1_5	LoUtente_2, GeUtente_2, LTitolo_2, FDurata_2, FAnno_1	Errore
TC_5.1_6	LoUtente_2, GeUtente_2, LTitolo_2, FDurata_2, FAnno_2, LDescrizione_1	Errore
TC_5.1_7	LoUtente_2, GeUtente_2, LTitolo_2, FDurata_2, FAnno_2, LDescrizione_2, CaCopertina_1	Errore
TC_5.1_8	LoUtente_2, GeUtente_2, LTitolo_2, FDurata_2, FAnno_2, LDescrizione_2, CaCopertina_2	Corretto

AggiungereArtista

PARAMETRO:	Utente
Loggato[LoUtente]:	1. L'utente che vuole aggiungere un artista non è loggato[Errore] 2. L'utente che vuole aggiungere un artista è loggato[Successo_LoUtente]
Gestore[GeUtente]:	1. L'utente che vuole aggiungere un artista non è gestore[if_LoUtente_OK] [Errore] 2. L'utente che vuole aggiungere un artista è gestore[if_GeUtente_OK] [Successo_Utente]

PARAMETRO:	Nome
------------	------

PARAMETRO:	Nome
Lunghezza[LNome]:	1. < 5 or > 100[Errore] 2. > 4 & < 101[Successo_LNome]

PARAMETRO:	Nascita
FORMATO:	^((19 20)\d{2})-(0[1-9] 1[0-2])-(0[1-9] [12]\d 3[01])\$
Formato[FNascita]:	1. Non rispetta il formato[Errore] 2. Rispetta il formato[Successo_FNascita]

PARAMETRO:	Descrizione
Lunghezza[LDescrizione]:	1. < 10[Errore] 2. > 9[Successo_LDescrizione]

PARAMETRO:	Faccia
FORMATO:	JPG or PNG
Caricamento[CaFaccia]:	1. Faccia non caricata[Errore] 2. Faccia caricata[Successo_CaFaccia]

Test cases

Codice	Combinazione	Esito
TC_5.2_1	LoUtente_1	Errore
TC_5.2_2	LoUtente_2, GeUtente_1	Errore
TC_5.2_3	LoUtente_2, GeUtente_2, LNome_1	Errore
TC_5.2_4	LoUtente_2, GeUtente_2, LNome_2, FNascita_1	Errore
TC_5.2_5	LoUtente_2, GeUtente_2, LNome_2, FNascita_2, LDescrizione_1	Errore
TC_5.2_6	LoUtente_2, GeUtente_2, LNome_2, FDurata_2, FAnno_2, LDescrizione_2, CaFaccia_1	Errore
TC_5.2_7	LoUtente_2, GeUtente_2, LNome_2, FDurata_2, FAnno_2, LDescrizione_2, CaFaccia_2	Corretto

AggiungereGenere

PARAMETRO:	Utente
Loggato[LoUtente]:	1. L'utente che vuole aggiungere un genere non è loggato[Errore] 2. L'utente che vuole aggiungere un genere è loggato[Successo_LoUtente]
Gestore[GeUtente]:	1. L'utente che vuole aggiungere un genere non è gestore[if_LoUtente_OK] [Errore] 2. L'utente che vuole aggiungere un genere è gestore[if_GeUtente_OK] [Successo_Utente]

PARAMETRO:	Nome
Lunghezza[LNome]:	1. < 3 or > 50[Errore] 2. > 3 & < 51[Successo_LNome]
Presenza[PNome]:	1. Genere già presente[if_LNome_OK][Errore] 2. Genere non presente[if_PNome_OK][Successo_Nome]

Test cases

Codice	Combinazione	Esito
TC_5.3_1	LoUtente_1	Errore
TC_5.3_2	LoUtente_2, GeUtente_1	Errore
TC_5.3_3	LoUtente_2, GeUtente_2, LNome_1	Errore
TC_5.3_4	LoUtente_2, GeUtente_2, LNome_2, PNome_1	Errore
TC_5.3_5	LoUtente_2, GeUtente_2, LNome_2, PNome_2	Corretto

ModificareGenere

PARAMETRO:	Utente
Loggato[LoUtente]:	1. L'utente che vuole modificare un genere non è loggato[Errore] 2. L'utente che vuole modificare un genere è loggato[Successo_LoUtente]
Gestore[GeUtente]:	1. L'utente che vuole modificare un genere non è gestore[if_LoUtente_OK][Errore] 2. L'utente che vuole modificare un genere è gestore[if_GeUtente_OK][Successo_Utente]

PARAMETRO:	Nome
Lunghezza[LNome]:	1. < 3 or > 50[Errore] 2. > 3 & < 51[Successo_LNome]
Presenza[PNome]:	1. Genere già presente[if_LNome_OK][Errore] 2. Genere non presente[if_PNome_OK][Successo_Nome]

Test cases

Codice	Combinazione	Esito
TC_5.4_1	LoUtente_1	Errore
TC_5.4_2	LoUtente_2, GeUtente_1	Errore
TC_5.4_3	LoUtente_2, GeUtente_2, LNome_1	Errore
TC_5.4_4	LoUtente_2, GeUtente_2, LNome_2, PNome_1	Errore
TC_5.4_5	LoUtente_2, GeUtente_2, LNome_2, PNome_2	Corretto

RimuovereGenere

PARAMETRO:	Utente
Loggato[LoUtente]:	1. L'utente che vuole rimuovere un genere non è loggato[Errore] 2. L'utente che vuole rimuovere un genere è loggato[Successo_LoUtente]
Gestore[GeUtente]:	1. L'utente che vuole rimuovere un genere non è gestore[if_LoUtente_OK][Errore] 2. L'utente che vuole rimuovere un genere è gestore[if_GeUtente_OK][Successo_Utente]

Test cases

Codice	Combinazione	Esito
TC_5.5_1	LoUtente_1	Errore
TC_5.5_2	LoUtente_2, GeUtente_1	Errore
TC_5.5_3	LoUtente_2, GeUtente_2	Corretto

AggiornareGeneriFilm

PARAMETRO:	Utente
Loggato[LoUtente]:	1. L'utente che vuole aggiornare generi ad un film non è loggato[Errore] 2. L'utente che vuole aggiornare generi ad un film è loggato[Successo_LoUtente]
Gestore[GeUtente]:	1. L'utente che vuole aggiornare generi ad un film non è gestore[if_LoUtente_OK][Errore] 2. L'utente che vuole aggiornare generi ad un film è gestore[if_GeUtente_OK][Successo_Utente]

Test cases

Codice	Combinazione	Esito
TC_5.6_1	LoUtente_1	Errore
TC_5.6_2	LoUtente_2, GeUtente_1	Errore
TC_5.6_3	LoUtente_2, GeUtente_2	Corretto