

1 Unit Testing

I test elencati di seguito sono stati eseguiti usando PostgreSQL.

1.1 Package Server.DAO

Test ID	Driver	UUT	Metodo	Obbiettivo	Precondizioni/Input	Comportamento atteso
1	AmministratoreDAOTest	AmministratoreDAO	save	Verificare che l'oggetto AmministratoreDAO venga correttamente salvato su database	Il database è vuoto	La funzione save() non deve restituire "false".
2	AmministratoreDAOTest	AmministratoreDAO	getAmministratoreById	Verificare che l'oggetto AmministratoreDAO venga correttamente salvato su database	È stato eseguito il test n°1. Alla funzione viene passato l'identificativo dell'oggetto AmministratoreDAO aggiunto al database dal precedente test.	La funzione deve restituire un oggetto AmministratoreDAO corrispondente a quello inserito nel precedente test.
3	AmministratoreDAOTest	AmministratoreDAO	getAmministratoreById	verificare che la funzione restituisca null se non esistono oggetti con l'identificativo specificato	Alla funzione viene passato un numero non corrispondente ad un identificativo di un oggetto AmministratoreDAO.	La funzione deve restituire un riferimento nullo.
4	AmministratoreDAOTest	AmministratoreDAO	update getAmministratoreById	Verificare che le modifiche eseguite su un oggetto vengano correttamente salvate su database.	Deve essere stato eseguito il Test n°1. Sull'oggetto usato per il suddetto test vengono operate modifiche.	La funzione update() non deve restituire "false", la funzione getAmministratoreById() deve restituire un oggetto i cui attributi rispecchino le modifiche effettuate.
5	AutoreDAOTest	AutoreDAO	save	verificare che l'oggetto sia salvato correttamente su database	Il database è vuoto	La funzione save() non deve restituire "false".
6	AutoreDAOTest	AutoreDAO	getAutoreById	verificare che l'oggetto sia salvato correttamente su database	È stato eseguito il test n°5. Alla funzione viene passato l'identificativo dell'oggetto AutoreDAO aggiunto al database dal precedente test.	La funzione deve restituire un oggetto AutoreDAO corrispondente a quello inserito nel precedente test.
7	AutoreDAOTest	AutoreDAO	getAutoreById	verificare che la funzione restituisca null se non esistono oggetti con l'identificativo specificato	Alla funzione viene passato un numero non corrispondente ad un identificativo di un oggetto AutoreDAO.	La funzione deve restituire un riferimento nullo.
8	AutoreDAOTest	AutoreDAO	verificaEsistenzaAutore	verificare che la funzione restituisca "true" se nel database è presente un autore il cui nome e cognome corrispondono a quelli specificati	Deve essere stato effettuato il test n°5. Alla funzione vengono passati nome e cognome dell'autore inserito nel database durante il test n° 5.	La funzione non deve restituire false.

9	AutoreDAOTest	AutoreDAO	verificaEsistenzaAutore	verificare che la funzione restituisca "false" se nel database non è presente un autore il cui nome e cognome corrispondono a quelli specificati	Alla funzione vengono passati nome e cognome diversi da quelli precedentemente inseriti	La funzione deve restituire false
10	AutoreDAOTest	AutoreDAO	update getAutoreById	Verificare che le modifiche eseguite su un oggetto vengano correttamente salvate su database.	Deve essere stato eseguito il Test n°5. Sull'oggetto usato per il suddetto test vengono operate modifiche.	La funzione update() non deve restituire "false", la funzione getAutoreById() deve restituire un oggetto i cui attributi rispecchino le modifiche effettuate.
11	AutoreDAOTest	AutoreDAO	getListaIdAutori	Verificare che la funzione restituisca una lista di identificativi.	Devono essere stati effettuati i test n°5 e n° 10.	La funzione deve restituire una lista contenente due elementi, corrispondenti agli identificativi degli oggetti inseriti nel database dai test n°5 e n° 10, ordinati per cognome, dunque 2 ed 1 in ordine.
12	TipologiaDAOTest	TipologiaDAO	save	Verificare che l'oggetto TipologiaDAO venga correttamente salvato su database	Il database è vuoto	La funzione save() non deve restituire "false".
13	TipologiaDAOTest	TipologiaDAO	save	Verificare che l'oggetto TipologiaDAO venga correttamente salvato su database se ad esso viene aggiunto un oggetto AutoreDAO		La funzione save() non deve restituire "false".
14	TipologiaDAOTest	TipologiaDAO	getTipologiaById	verificare che l'oggetto sia salvato correttamente su database	È stato eseguito il test n°12. Alla funzione viene passato l'identificativo dell'oggetto TipologiaDAO aggiunto al database dal precedente test.	La funzione deve restituire un oggetto TipologiaDAO corrispondente a quello inserito nel precedente test.
15	TipologiaDAOTest	TipologiaDAO	getTipologiaById	verificare che l'oggetto sia salvato correttamente su database, assieme alla sua lista di oggetti AutoreDAO	È stato eseguito il test n°13. Alla funzione viene passato l'identificativo dell'oggetto TipologiaDAO aggiunto al database dal precedente test.	La funzione deve restituire un oggetto TipologiaDAO corrispondente a quello inserito nel precedente test.
16	TipologiaDAOTest	TipologiaDAO	getTipologiaById	verificare che la funzione restituisca null se non esistono oggetti con l'identificativo specificato	Alla funzione viene passato un numero non corrispondente ad un identificativo di un oggetto TipologiaDAO.	La funzione deve restituire un riferimento nullo.

17	TipologiaDAOTest	TipologiaDAO	update getTipologiaById	Verificare che le modifiche eseguite su un oggetto vengano correttamente salvate su database.		La funzione update() non deve restituire "false", la funzione getAutoreById() deve restituire un oggetto i cui attributi rispecchino le modifiche effettuate.
18	TipologiaDAOTest	TipologiaDAO	getListIdTipologie	Verificare che la funzione restituisca una lista di identificativi.	Devono essere stati effettuati i test n°12, n°13 e n° 17.	La funzione deve restituire una lista contenente tre elementi, corrispondenti agli identificativi degli oggetti inseriti nel database dai test n°12, n°13 e n° 17.
19	RisorsaDAOTest	RisorsaDAO	save	Verificare che l'oggetto RisorsaDAO venga correttamente salvato su database	Il database è vuoto	La funzione save() non deve restituire "false".
20	RisorsaDAOTest	RisorsaDAO	getRisorsaById	verificare che l'oggetto sia salvato correttamente su database	È stato eseguito il test n°19. Alla funzione viene passato l'identificativo dell'oggetto RisorsaDAO aggiunto al database dal precedente test.	La funzione deve restituire un oggetto RisorsaDAO corrispondente a quello inserito nel precedente test.
21	RisorsaDAOTest	RisorsaDAO	getRisorsaById	verificare che la funzione restituisca null se non esistono oggetti con l'identificativo specificato	Alla funzione viene passato un numero non corrispondente ad un identificativo di un oggetto RisorsaDAO.	La funzione deve restituire un riferimento nullo.
22	RisorsaDAOTest	RisorsaDAO	verificaDuplicato	verificare che la funzione restituisca "false" se non esiste un oggetto RisorsaDAO avente titolo ed autore specificati	Viene creato un autore ad-hoc per il test, dopo di che la funzione viene invocata due volte: una prima volta con un autore cui non corrisponde nessuna risorsa, la seconda con un titolo cui non corrisponde nessuna risorsa.	la funzione deve restituire false in entrambi i casi.
23	RisorsaDAOTest	RisorsaDAO	verificaDuplicato	verificare che la funzione restituisca "true" se esiste un oggetto RisorsaDAO avente titolo ed autore specificati	Deve essere stato eseguito il test n° 19. La funzione viene stimolata con titolo ed autore relativi all'oggetto RisorsaDAO inserito nel database dal test sopracitato.	La funzione deve restituire true.
24	RisorsaDAOTest	RisorsaDAO	getListRisorse	Verificare che la funzione restituisca una lista di identificativi.	Deve essere stato effettuato il test n° 19.	La funzione restituisce una lista contenente un solo elemento, corrispondente all'identificativo dell'oggetto RisorsaDAO inserito nel database durante il test n°19.

25	UtenteDAOTest	UtenteDAO	save	Verificare che l'oggetto UtenteDAO venga correttamente salvato su database	Il database è vuoto	La funzione save() non deve restituire "false".
26	UtenteDAOTest	UtenteDAO	getUtenteById	verificare che l'oggetto sia salvato correttamente su database	È stato eseguito il test n°25. Alla funzione viene passato l'identificativo dell'oggetto UtenteDAO aggiunto al database dal precedente test.	La funzione deve restituire un oggetto UtenteDAO corrispondente a quello inserito nel precedente test.
27	UtenteDAOTest	UtenteDAO	getUtenteById	verificare che la funzione restituisca null se non esistono oggetti con l'identificativo specificato	Alla funzione viene passato un numero non corrispondente ad un identificativo di un oggetto UtenteDAO.	La funzione deve restituire un riferimento nullo.
28	UtenteDAOTest	PreferenzaUtenteDAO	save	Verificare che l'oggetto PreferenzaUtenteDAO venga correttamente salvato su database		La funzione save() non deve restituire "false".
29	UtenteDAOTest	PreferenzaUtenteDAO	getListaPreferenzeUtente	Verificare che l'oggetto PreferenzaUtenteDAO venga correttamente salvato su database	Deve esser stato eseguito il test n°28. Viene chiesta la lista delle preferenze per l'utente alla quale queste sono state aggiunte, nel test sopracitato.	La funzione deve restituire una lista contenente esattamente due elementi. Viene verificato che essi corrispondano agli oggetti PreferenzaUtenteDAO specificati al test precedente.
30	UtenteDAOTest	UtenteDAO	update getUtenteById getListaPreferenze	Verificare che l'oggetto PreferenzaUtenteDAO venga correttamente salvato su database	Deve esser stato eseguito il test n°28. Viene chiesta la lista delle preferenze per l'utente alla quale queste sono state aggiunte, nel test n°28.	La funzione deve restituire una lista contenente esattamente due elementi. Viene verificato che essi corrispondano agli oggetti PreferenzaUtenteDAO specificati al test n°28.
31	CatalogoUtenteDAOTest	CatalogoUtenteDAO	save update	Verificare che l'oggetto CatalogoUtenteDAO e l'oggetto UtenteDAO cui esso è associato vengano correttamente salvati su database	Il database è vuoto	La funzione save() deve restituire true. La funzione update deve restituire true.
32	CatalogoUtenteDAOTest	CatalogoUtenteDAO	getCatalogoUtenteById update	Verificare che l'oggetto CatalogoUtenteDAO e l'oggetto UtenteDAO cui esso è associato vengano correttamente salvati su database.	Deve esser stato eseguito il test n°31. Alla funzione getCatalogoUtenteById viene passato l'identificativo dell'oggetto CatalogoUtenteDAO aggiunto al database dal precedente test. Vengono aggiunti oggetti RisorsaDAO all'oggetto, che viene aggiornato.	la funzione getCatalogoUtenteById non deve restituire un riferimento nullo, la funzione update deve restituire true.

33	CatalogoUtenteDAOTest	CatalogoUtenteDAO	getCatalogoUtenteById	Verificare che l'oggetto CatalogoUtenteDAO e l'oggetto UtenteDAO cui esso è associato vengano correttamente salvati su database.	Deve esser stato eseguito il test n°32. Alla funzione getCatalogoUtenteById viene passato l'identificativo dell'oggetto CatalogoUtenteDAO aggiunto al database dal test n°31.	Viene controllato che l'oggetto CatalogoUtenteDAO appaia come ci si aspetta. l'identificativo dell'oggetto UtenteDAO corrisponde a quello impostato nel test n°31, gli identificativi delle risorse corrispondono a quelli inseriti nel test n°32.
34	CatalogoUtenteDAOTest	CatalogoUtenteDAO	getCatalogoUtenteById	verificare che la funzione restituisca null se non esistono oggetti con l'identificativo specificato	Alla funzione viene passato un numero non corrispondente ad un identificativo di un oggetto CatalogoUtenteDAO.	La funzione deve restituire un riferimento nullo.
35	CatalogoUtenteDAOTest	CatalogoUtenteDAO	getListIdCataloghi	Verificare che la funzione restituisca una lista di identificativi.	Deve essere stato eseguito il test n° 31.	Viene verificato che la funzione restituisca una lista di un solo elemento, corrispondente all'identificativo dell'oggetto CatalogoUtenteDAO inserito nel database dal test n°31.
36	DataAcquistoDAOTest	DataAcquistoDAO	save	Verificare che l'oggetto DataAcquistoDAO venga correttamente salvato su database	Il database è vuoto. Viene creato un oggetto RisorsaDAO ed un oggetto CatalogoUtenteDAO con cui settare gli attributi dell'oggetto DataAcquistoDAO. Poi lo si salva su database.	La funzione save deve restituire true.
37	DataAcquistoDAOTest	DataAcquistoDAO	getDateAcquistoPerMese	Verificare che l'oggetto DataAcquistoDAO venga correttamente salvato su database. Verificare che la funzione getDateAcquistoPerMese restituisca il numero corretto di oggetti DataAcquistoDAO	Vengono inseriti su database alcuni oggetti DataAcquistoDAO, aventi date di acquisto 18/02/2017, 18/02/2018, 18/01/2018. Viene chiamata la funzione getDateAcquistoPerMese(2).	La funzione deve restituire un solo elemento DataAcquistoDAO, corrispondente all'acquisto effettuato il 18/02/2018.
38	DataAcquistoDAOTest	DataAcquistoDAO	getDateAcquistoPerMese	Verificare che l'oggetto DataAcquistoDAO venga correttamente salvato su database. Verificare che la funzione getDateAcquistoPerMese restituisca il numero corretto di oggetti DataAcquistoDAO	Deve essere stato eseguito il test n°37. Viene chiamata la funzione getDateAcquistoPerMese(3).	La lista restituita dalla funzione deve essere vuota.

39	DataAcquistoDAOTest	DataAcquistoDAO	getDataAcquistoAnnoSolare	Verificare che l'oggetto DataAcquistoDAO venga correttamente salvato su database. Verificare che la funzione getDataAcquistoAnnoSolare restituisca il numero corretto di oggetti DataAcquistoDAO	Deve essere stato eseguito il test n°37. Alla funzione viene passato il catalogo aggiornato proprio durante quel test.	La lista restituita dalla funzione deve contenere esattamente due elementi, corrispondenti agli acquisti effettuati il 18/02/2018 ed il 18/01/2018.
40	DataNoleggioDAOTest	DataNoleggioDAO	save	Verificare che l'oggetto DataNoleggioDAO venga correttamente salvato su database	Il database è vuoto. Viene creato un oggetto RisorsaDAO ed un oggetto CatalogoUtenteDAO con cui settare gli attributi dell'oggetto DataNoleggioDAO. Poi lo si salva su database.	La funzione save deve restituire true.
41	DataNoleggioDAOTest	DataNoleggioDAO	getDateNoleggioPerMese	Verificare che l'oggetto DataNoleggioDAO venga correttamente salvato su database. Verificare che la funzione getDateNoleggioPerMese restituisca il numero corretto di oggetti DataNoleggioDAO	Vengono inseriti su database alcuni oggetti DataNoleggioDAO, aventi date di acquisto 18/02/2017, 18/02/2018, 18/01/2018. Viene chiamata la funzione getDateNoleggioPerMese(2).	La funzione deve restituire un solo elemento DataNoleggioDAO, corrispondente all'acquisto effettuato il 18/02/2018.
42	DataNoleggioDAOTest	DataNoleggioDAO	getDateNoleggioPerMese	Verificare che l'oggetto DataNoleggioDAO venga correttamente salvato su database. Verificare che la funzione getDateNoleggioPerMese restituisca il numero corretto di oggetti DataNoleggioDAO	Deve essere stato eseguito il test n°41. Viene chiamata la funzione getDateNoleggioPerMese(3).	La lista restituita dalla funzione deve essere vuota.
43	DAONullPtrTest	AmministratoreDAO	getAmministratoreById	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La funzione deve restituire null
44	DAONullPtrTest	AutoreDAO	getAutoreById	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La funzione deve restituire null
45	DAONullPtrTest	AutoreDAO	verificaEsistenzaAutore	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	la funzione deve restituire false
46	DAONullPtrTest	TipologiaDAO	getTipologiaById	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La funzione deve restituire null

47	DAONullPtrTest	RisorsaDAO	getRisorsaById	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La funzione deve restituire null
48	DAONullPtrTest	RisorsaDAO	verificaDuplicato	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La funzione deve restituire "true", di modo che l'inserimento non possa essere effettuato
49	DAONullPtrTest	UtenteDAO	getUtenteById	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La funzione deve restituire null
50	DAONullPtrTest	CatalogoUtenteDAO	getCatalogoUtenteById	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La funzione deve restituire null
51	DAONullPtrTest	DataAcquistoDAO	getDataAcquistoPerMese	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La lista restituita dalla funzione deve essere vuota.
52	DAONullPtrTest	DataAcquistoDAO	GetDataAcquistoAnnoSolare	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La lista restituita dalla funzione deve essere vuota.
53	DAONullPtrTest	DataNoleggioDAO	getDataNoleggioPerMese	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La lista restituita dalla funzione deve essere vuota.

1.2 Package Server.Entity

I test sul package Entity sono stati effettuati usando dati fittizi, riportati in appendice. Tali dati vengono inseriti nel database prima di ogni test, dopo averlo ripulito dagli effetti dei test precedenti. L'inizializzazione avviene usando i comandi SQL contenuti nello script initDatabase.sql, presente nella directory "database".

1.2.1 Elenco dei test

Test ID	Driver	UUT	Metodo	Obbiettivo	Precondizioni/Input	Comportamento atteso
A01	AutoreTest	ListaAutori	getListaAutori	verificare che gli autori presenti nel database siano correttamente prelevati ed ordinati per cognome autore.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql.	La funzione deve restituire una lista contenente 10 elementi, i cui identificativi sono, in ordine, 52, 12, 8, 10, 11, 5, 6, 9, 7, 13.
A02	AutoreTest	ListaAutori	getAutoreById	verificare che la funzione getAutoreById restituisca un riferimento nullo se nel database ne è presente un oggetto con id corrispondente a quello specificato.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. Viene chiesto l'Autore di id = 14.	la funzione deve restituire null
A03	AutoreTest	ListaAutori	getAutoreById	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La funzione deve restituire null
A04	AutoreTest	ListaAutori Autore	getAutoreById getNome getCognome	verificare che la funzione getAutoreById restituisca un oggetto correttamente, se nel database ne è presente uno con id corrispondente.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. Alla funzione viene passato id=5.	La funzione getAutoreById non deve restituire null, ma un oggetto Autore i cui campi nome e cognome siano, rispettivamente, "Ennio" e "Morricone"
A05	AutoreTest	Autore ListaAutori	addAutore save getAutoreById	Con la funzione addAutore viene creato un nuovo autore, poi salvato nel database. getAutoreById viene usata, per prelevare l'oggetto dal database e controllare che sia stato salvato correttamente.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql.	Al nuovo oggetto Autore viene associato id=1. Tale id viene usato per prelevare l'oggetto dal database, confrontando i valori con quelli che ci si aspetta.
T01	TipologiaTest	ListaTipologie	getListaTipologie	verificare che le tipologie presenti nel database siano correttamente prelevate	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql.	La funzione deve restituire una lista di esattamente quattro elementi, i cui identificativi sono, in ordine, 1, 2, 3, 4
T02	TipologiaTest	ListaTipologie	getTipologiaById	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La funzione deve restituire null

T03	TipologiaTest	ListaTipologie	getTipologiaById	verificare che la funzione restituisca un riferimento nullo se nel database ne è presente un oggetto con id corrispondente a quello specificato.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. La funzione viene sollecitata con id=9999, che è inesistente.	la funzione deve restituire un riferimento nullo
T04	TipologiaTest	ListaTipologie Tipologia	getTipologiaById getDescrizione getListaAutori	verificare che la getTipologiaById restituisca un oggetto correttamente, se nel database ne è presente uno con id corrispondente.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. Alla funzione viene passato id=1, corrispondente alla tipologia "Audio", la quale possiede riferimenti con gli Autori di identificativo pari a 5, 6 e 7.	La funzione getTipologiaById non deve restituire null, ma un oggetto Tipologia il cui campo descrizione sia "Audio". La funzione getListaAutori deve restituire una lista di esattamente tre elementi, corrispondenti agli autori di id pari a 5, 6 e 7.
T05	TipologiaTest	ListaTipologie Tipologia	getTipologiaById update getListaAutori	verificare che le modifiche operate su un oggetto Tipologia siano correttamente rese permanenti.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. Alla tipologia viene aggiunto un nuovo autore, quello di id=8, chiamando update per salvare le modifiche. Usando getTipologiaById viene verificato che esse siano state correttamente rese permanenti.	La funzione getListaAutori deve restituire una lista di esattamente quattro elementi, corrispondenti agli autori di id pari a 5, 6, 7 ed 8.
R01	RisorsaTest	CatalogoMultimediale	getListaRisorse	verificare che l'elenco di tutte le risorse disponibili sia correttamente prelevato	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql.	La funzione deve restituire una lista di esattamente 10 elementi, i cui identificativi sono, in ordine, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23 e 53
R02	RisorsaTest	CatalogoMultimediale	getRisorsaById	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La funzione deve restituire null
R03	RisorsaTest	CatalogoMultimediale	getRisorsaById	verificare che la funzione restituisca un riferimento nullo se nel database ne è presente un oggetto con id corrispondente a quello specificato.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. La funzione viene sollecitata con id=9999, che è inesistente.	la funzione deve restituire un riferimento nullo
R04	RisorsaTest	CatalogoMultimediale Risorsa	getRisorsaById getTipologia getAutore	verificare che la getRisorsaById restituisca un oggetto correttamente, se nel database ne è presente uno con id corrispondente. Verifica che gli oggetti Tipologia ed Autore associati all'oggetto risorsa siano correttamente prelevati.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. Alla funzione getRisorsaById viene passato id=18, a cui corrisponde un oggetto nel database.	la funzione getRisorsaById deve restituire un oggetto avente le seguenti caratteristiche: titolo='The Aviator', costoacquisto=9, costonoleggio=2, datapubblicazione='2018-02-22', idautore=9, idtipologia=2

R05	RisorsaTest	CatalogoMultimediale Risorsa	inserisciRisorsa getRisorsaById save	verificare che un oggetto Risorsa sia reso persistente in modo corretto.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql.	Viene creato un nuovo oggetto Risorsa avente le seguenti caratteristiche: titolo='Goodfellas', costoacquisto=9, costonoleggio=2, datapubblicazione='2018-02-22', idautore=9, idtipologia=2, a cui verrà associato id=1. La funzione getRisorsaById verrà usata per verificare che tale oggetto sia stato reso persistente in modo corretto.
R06	RisorsaTest	CatalogoMultimediale	inserisciRisorsa	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La funzione deve restituire false e nessun elemento deve essere aggiunto sul database.
R07	RisorsaTest	CatalogoMultimediale	verificaDuplicato	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La funzione deve restituire "true", di modo che l'inserimento non possa essere effettuato
R08	RisorsaTest	CatalogoMultimediale	verificaDuplicato	verificare che la funzione restituisca "false" se non esiste un oggetto Risorsa avente titolo ed autore specificati	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. La funzione viene sollecitata con titolo="Bohemian Rhapsody", autore il cui id è 5;	la funzione deve restituire false
R09	RisorsaTest	CatalogoMultimediale	verificaDuplicato	verificare che la funzione restituisca "true" se esiste un oggetto Risorsa avente titolo ed autore specificati	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. La funzione viene sollecitata con titolo='The Shining', autore il cui id è 8.	La funzione deve restituire true.
U01		ListaCataloghiUtenti	getListaCatalogoUtenti	verificare che i cataloghi presenti nel database siano correttamente prelevati	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql.	La funzione deve restituire una lista contenente esattamente tre elementi, i cui identificativi sono, in ordine, 27, 28 e 29
U02		ListaCataloghiUtenti	getUtenteById	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La funzione deve restituire null
U03		ListaCataloghiUtenti	getUtenteById	verificare che la funzione restituisca un riferimento nullo se nel database ne è presente un oggetto con id corrispondente a quello specificato.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. La funzione viene sollecitata con id=9999, che è inesistente.	la funzione deve restituire un riferimento nullo

U04		ListaCataloghiUtenti Utente	getUtenteById getNome getCognome getEmail getPassword	verificare che la getUtenteById restituisca un oggetto correttamente, se nel database ne è presente uno con id corrispondente. I test sul fetch del catalogo utente e delle sue preferenze saranno effettuati in un secondo momento	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. Alla funzione viene passato id=24	La funzione getUtenteById non deve restituire null, ma un oggetto Utente, caratterizzato come segue: nome='Salvatore' cognome='Barone' email='salvatorebarone@email.com' password='passw0rd.
U05		ListaCataloghiUtenti	getCatalogoUtenteById	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La funzione deve restituire null
U05		ListaCataloghiUtenti	getCatalogoUtenteById	verificare che la funzione restituisca un riferimento nullo se nel database ne è presente un oggetto con id corrispondente a quello specificato.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. La funzione viene sollecitata con id=9999, che è inesistente.	la funzione deve restituire un riferimento nullo
U07		ListaCataloghiUtenti CatalogoUtente	getCatalogoUtenteById getUtente getListeRisorsePossedute	verificare che la getCatalogoUtenteById restituisca un oggetto correttamente, se nel database ne è presente uno con id corrispondente.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. Alla funzione viene passato id=27.	La funzione deve restituire un riferimento ad un oggetto CatalogoUtente valido; getUtente deve restituire l'oggetto Utente avente id 24 e getListeRisorsePossedute deve restituire una lista contenente esattamente due oggetti Risorsa, aventi id 18 e 23, rispettivamente
U08		ListaCataloghiUtenti CatalogoUtente	getCatalogoUtenteById addRisorsa update	verificare che le modifiche operate su un oggetto Tipologia siano correttamente rese permanenti.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. All'oggetto CatalogoUtente con id=27 viene aggiunto l'oggetto Risorsa avente id=18, poi viene chiamata la funzione update.	getListeRisorsePossedute deve restituire una lista contenente esattamente tre oggetti Risorsa, aventi id 17, 18 e 23, rispettivamente
U09		ListaCataloghiUtenti CatalogoUtente	getCatalogoUtenteById addRisorsa	Verificare il comportamento della funzione quando uno o più parametri sono null.	Alla funzione addRisorsa viene passato un puntatore nullo.	Nessuna modifica deve essere effettuata sull'insieme delle risorse appartenenti al catalogo.
U10		Utente	getCatalogo	verificare che l'oggetto CatalogoUtente associato all'oggetto Utente sia correttamente caricato	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. Alla funzione getUtenteById viene passato id=24.	La funzione getCatalogo deve restituire un oggetto CatalogoUtente avente id=27, getListeRisorsePossedute, chiamata su tale oggetto CatalogoUtente, deve restituire una lista contenente esattamente tre oggetti Risorsa, aventi id 17, 18 e 23, rispettivamente

P01	PreferenzeTest	ListaPreferenzeUtente	getListaPreferenzeUtente	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La lista restituita dalla funzione deve essere vuota.
P02	PreferenzeTest	ListaPreferenzeUtente	getListaPreferenzeUtente	verificare che le preferenze utente siano correttamente prelevate dal database	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. Alla funzione viene passato il catalogo il cui id è 27.	La funzione deve restituire una lista contenente esattamente due elementi, contenenti, rispettivamente, le seguenti preferenze: {autore=7, tipologia=1} ed {autore=12, tipologia=3}
P03	PreferenzeTest	ListaPreferenzeUtente	aggiungiPreferenza	verificare che le preferenze di un utente siano rese persistenti in modo corretto.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. Al catalogo il cui id è 27 viene aggiunta la seguente preferenza {autore=5, tipologia=1}	La funzione getListaPreferenzeUtente deve restituire una lista contenente esattamente tre elementi, contenenti, rispettivamente, le seguenti preferenze: {autore=7, tipologia=1}, {autore=12, tipologia=3} ed {autore=5, tipologia=1}
P04	PreferenzeTest	ListaPreferenzeUtente	aggiungiPreferenza	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La funzione deve restituire false e nessuna preferenza deve essere aggiunta all'utente.
S01	StoricoIncassiTest	StoricoIncassi	getDataAcquistoAnnoSolare	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La lista restituita dalla funzione deve essere vuota.
S02	StoricoIncassiTest	StoricoIncassi	getListaDateAcquistoAnnoSolare	verificare che l'elenco degli acquisti venga correttamente prelevato	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. La funzione viene stimolata con un oggetto catalogo avente identificativo pari a 27.	La funzione deve restituire una lista contenente esattamente un elemento, caratterizzato da dataAcquisto=22/02/2018 idRisorsa=23
S03	StoricoIncassiTest	StoricoIncassi	getListaDateAcquistoPerMese	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La lista restituita dalla funzione deve essere vuota.
S04	StoricoIncassiTest	StoricoIncassi	getListaDateAcquistoPerMese	verificare che l'elenco degli acquisti per mese venga correttamente prelevato	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. La funzione viene stimolata con un oggetto catalogo avente identificativo pari a 27, mese=2	La funzione deve restituire una lista contenente esattamente quattro elementi, caratterizzato da {dataAcquisto=22/02/2018 idRisorsa=23}, {dataAcquisto=22/02/2018 idRisorsa=15}, {dataAcquisto=22/02/2018 idRisorsa=19}, {dataAcquisto=22/02/2018 idRisorsa=53}

S05	StoricoIncassiTest	StoricoIncassi	getListaDateAcquistoPerMese	verificare che l'elenco degli acquisti per mese venga correttamente prelevato	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. La funzione viene stimolata con un oggetto catalogo avente identificativo pari a 27, mese=3	La funzione deve restituire una lista vuota.
S06	StoricoIncassiTest	StoricoIncassi	getListaDateNoleggioPerMese	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La lista restituita dalla funzione deve essere vuota.
S07	StoricoIncassiTest	StoricoIncassi	getListaDateNoleggioPerMese	verificare che l'elenco dei noleggi per mese venga correttamente prelevato	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. La funzione viene stimolata con un oggetto catalogo avente identificativo pari a 27, mese=2	La funzione deve restituire una lista contenente esattamente tre elementi, caratterizzato da {dataNoleggio=22/02/2018 idRisorsa=21}, {dataNoleggio=22/02/2018 idRisorsa=20}, {dataNoleggio=22/02/2018 idRisorsa=23}
S08	StoricoIncassiTest	StoricoIncassi	getListaDateNoleggioPerMese	verificare che l'elenco dei noleggi per mese venga correttamente prelevato	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. La funzione viene stimolata con un oggetto catalogo avente identificativo pari a 27, mese=3	La funzione deve restituire una lista vuota.

1.3 Package Server.BusinessLogic

Test ID	Driver	UUT	Metodo	Obbiettivo	Precondizioni/Input	Comportamento atteso
GR01	GestoreReportTest	GestoreReport	getReport	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La funzione deve restituire zero
GR02	GestoreReportTest	GestoreReport	getReport	Verificare che il totale degli acquisti e dei noleggi, per un mese specifico, delle risorse appartenenti ad una determinata tipologia venga calcolato correttamente.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. La funzione viene usata per calcolare il totale di acquisti e noleggi di risorse di tipo 2 (Video) nel mese di febbraio (2).	Il totale deve essere 7.3
GR03	GestoreReportTest	GestoreReport	getReport	Verificare che il totale degli acquisti e dei noleggi, per un mese specifico, delle risorse appartenenti ad una determinata tipologia venga calcolato correttamente.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. La funzione viene usata per calcolare il totale di acquisti e noleggi di risorse di tipo 2 (Video) nel mese di gennaio (1).	Il totale deve essere 0
GR04	GestoreReportTest	GestoreReport	getReport	Verificare il comportamento della funzione quando il parametro passato alla funzione è un numero non compreso nell'intervallo [1, 12]	La funzione viene chiamata con mese = 0 e mese = 13.	Il totale deve essere 0
GI01	GestoreIncassiTest	GestoreIncassi	getListaUtentiPremium	verificare che la lista degli utenti premium (utenti che abbiano effettuato acquisti per più di 100€ nell'ultimo anno solare) sia correttamente generata	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql.	La funzione deve restituire una lista contenente un solo oggetto CatalogoUtente, avente id = 29.
GP01	GestorePreferenzeTest	GestorePreferenze	getListaPreferenzeUtente	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La lista restituita dalla funzione deve essere vuota.
GP02	GestorePreferenzeTest	GestorePreferenze	getListaPreferenzeUtente	verificare che le preferenze utente siano correttamente prelevate dal database	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. Alla funzione viene passato il catalogo il cui id è 27.	La funzione deve restituire una lista contenente esattamente due elementi, contenenti, rispettivamente, le seguenti preferenze: {autore=7, tipologia=1} ed {autore=12, tipologia=3}

GP03	GestorePreferenzeTest	GestorePreferenze	aggiungiPreferenza	verificare che le preferenze di un utente siano rese persistenti in modo corretto.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. Al catalogo il cui id è 27 viene aggiunta la seguente preferenza {autore=5, tipologia=1}	La funzione getListePreferenzeUtente deve restituire una lista contenente esattamente tre elementi, contenenti, rispettivamente, le seguenti preferenze: {autore=7, tipologia=1}, {autore=12, tipologia=3} ed {autore=5, tipologia=1}
GP04	GestorePreferenzeTest	GestorePreferenze	aggiungiPreferenza	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La funzione deve restituire false e nessuna preferenza deve essere aggiunta all'utente.
GC01	GestoreCatalogoTest	GestoreCatalogo	getListeAutori	verificare che gli autori presenti nel database siano correttamente prelevati ed ordinati per cognome autore.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql.	La funzione deve restituire una lista contenente 10 elementi, i cui identificativi sono, in ordine, 52, 12, 8, 10, 11, 5, 6, 9, 7, 13.
GC02	GestoreCatalogoTest	GestoreCatalogo	getListeTipologie	verificare che le tipologie presenti nel database siano correttamente prelevate	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql.	La funzione deve restituire una lista di esattamente quattro elementi, i cui identificativi sono, in ordine, 1, 2, 3, 4
GC03	GestoreCatalogoTest	GestoreCatalogo	getListeRisorseNonPossedute	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La lista restituita dalla funzione deve essere vuota.
GC04	GestoreCatalogoTest	GestoreCatalogo	getListeRisorseNonPossedute	Verificare che la lista delle risorse non appartenenti ad un catalogo utente sia correttamente generata.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. Alla funzione viene chiesto l'elenco delle risorse non appartenenti al catalogo di id=27.	La funzione deve generare una lista di esattamente otto elementi, contenente le Risorse il cui identificativo corrisponde a 14, 15, 16, 17, 19, 20, 21 e 22
GC05	GestoreCatalogoTest	GestoreCatalogo	addOfferta	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La funzione deve restituire false
GC06	GestoreCatalogoTest	GestoreCatalogo	addOfferta	Verificare che le risorse offerte ad un utente siano correttamente salvate nel suo catalogo.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. Le risorse di id pari a 14, 15 e 16 vengono aggiunte al catalogo di id 27.	Chiamando la funzione CatalogoUtente::getListeRisorsePossedute deve essere restituita una lista contenente esattamente 5 elementi, ossia le risorse di id 23, 18, 14, 15 e 16.
GC07	GestoreCatalogoTest	GestoreCatalogo	inserisciRisorsa	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La funzione deve restituire false

GC08	GestoreCatalogoTest	GestoreCatalogo	inserisciRisorsa	verificare che un oggetto Risorsa sia reso persistente in modo corretto.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql.	Viene creato un nuovo oggetto Risorsa avente le seguenti caratteristiche: titolo='Goodfellas', costoacquisto=9, costonoleggio=2, datapubblicazione='2018-02-22', idautore=9, idtipologia=2, a cui verrà associato id=1. La funzione getRisorsaById verrà usata per verificare che tale oggetto sia stato reso persistente in modo corretto.
GC09	GestoreCatalogoTest	GestoreCatalogo	inserisciRisorsa	Verificare che, all'inserimento di una risorsa, la relazione tra tipologia risorsa ed autore venga creata correttamente, se non esiste.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. Viene inserita una nuova risorsa avente tipologia=1, autore = 52.	Viene verificato che la dimensione della lista autori per la tipologia 1 passi da 3 a 4 elementi a seguito dell'inserimento.
GC10	GestoreCatalogoTest	GestoreCatalogo	inserisciRisorsa	Verificare che, all'inserimento di una risorsa, la relazione tra tipologia risorsa ed autore non venga duplicata, se già esistente.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. Viene inserita una nuova risorsa avente tipologia=1, autore = 52.	Viene verificato che la dimensione della lista autori per la tipologia 1 sia 4 elementi, e che resti di 4 elementi a seguito dell'inserimento.
GC11	GestoreCatalogoTest	GestoreCatalogo	aggiungiAutore	Verificare che un autore venga correttamente inserito e reso persistente.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. Viene aggiunto un nuovo oggetto autore al database. L'id che gli viene assegnato è 1 (se il db è gestito da PostgreSQL). L'oggetto viene prelevato da database per verificare che sia stato correttamente salvato.	Al nuovo oggetto Autore viene associato id=4. Tale id viene usato per prelevare l'oggetto dal database, confrontando i valori con quelli che ci si aspetta.
GC12	GestoreCatalogoTest	GestoreCatalogo	aggiungiAutore	verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Uno o più parametri della funzione sono riferimenti nulli	La funzione deve restituire false

1.4 Package Server.Coordinator

Test ID	Driver	UUT	Metodo	Obbiettivo	Precondizioni/Input	Comportamento atteso
CU01	CoordinatoreUtenteTest	CoordinatoreUtente	getListaTipologiaRisorse	Verificare che la lista delle tipologie sia correttamente prelevata e restituita sottoforma di stringhe.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql.	La funzione deve restituire una lista composta da esattamente quattro elementi, in ordine: - "Audio" - "Video" - "Immagine" - "Podcast"
CU02	CoordinatoreUtenteTest	CoordinatoreUtente	getListaAutoriPerTipologia	Verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql. Il parametro della funzione è posto a null	Deve essere restituita una lista vuota
CU03	CoordinatoreUtenteTest	CoordinatoreUtente	getListaAutoriPerTipologia	Verificare il comportamento della funzione quando essa viene chiamata prima di getListaAutoriPerTipologia, quindi prima che siano disponibili le informazioni circa le tipologie gestite.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql. La funzione viene chiamata prima di aver chiamato getListaTipologiaRisorse.	Deve essere restituita una lista vuota
CU04	CoordinatoreUtenteTest	CoordinatoreUtente	getListaAutoriPerTipologia	Verificare il comportamento della funzione quando essa viene chiamata con un parametro il cui valore è errato.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql. La funzione viene chiamata dopo di aver chiamato getListaTipologiaRisorse, ma con una stringa diversa da quelle contenute dalla lista restituita da quest'ultima.	Deve essere restituita una lista vuota
CU05	CoordinatoreUtenteTest	CoordinatoreUtente	getListaAutoriPerTipologia	Verificare il comportamento della funzione quando essa viene chiamata con un parametro il cui valore è corretto.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql. La funzione viene chiamata dopo di aver chiamato getListaTipologiaRisorse, con tipologia="Audio"	Deve essere restituita una lista di esattamente tre elementi, in ordine: - "Morricone Ennio" - "Mozart W. A." - "Van Beethoven Ludwig"
CU06	CoordinatoreUtenteTest	CoordinatoreUtente	inserisciPreferenza	Verificare il comportamento della funzione quando uno o più parametri sono riferimenti nulli	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql. Uno o più parametri della funzione sono posti a null	Deve essere restituito il valore booleano false.

CU07	CoordinatoreUtenteTest	CoordinatoreUtente	inserisciPreferenza	Verificare il comportamento della funzione quando essa viene chiamata prima che siano disponibili le informazioni relative alle tipologie	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql. La funzione viene chiamata prima che siano disponibili le informazioni relative alle tipologie, quindi prima che sia stata chiamata la funzione getListAutoriPerTipologia	Deve essere restituito il valore booleano false.
CU08	CoordinatoreUtenteTest	CoordinatoreUtente	inserisciPreferenza	Verificare il comportamento della funzione quando essa viene chiamata prima che siano disponibili le informazioni relative agli autori per una certa tipologia	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql. La funzione viene chiamata prima che siano disponibili le informazioni relative alle tipologie, quindi prima che sia stata chiamata la funzione getListAutoriPerTipologia	Deve essere restituito il valore booleano false.
CU09	CoordinatoreUtenteTest	CoordinatoreUtente	inserisciPreferenza	Verificare il comportamento della funzione quando essa viene chiamata con un parametro il cui valore è errato.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql. Il parametro "utente" ha un valore diverso da quelli consentiti, oppure il parametro autore ha un valore diverso da quelli presenti nella lista restituita da getListAutoriPerTipologia, oppure il parametro tipologia ha un valore diverso da quelli presenti nella lista restituita da getListTipologiaRisorse.	Deve essere restituito il valore booleano false.

CU10	CoordinatoreUtenteTest	CoordinatoreUtente	inserisciPreferenza	Verificare il comportamento della funzione quando essa viene chiamata con un parametro il cui valore è corretto.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql. Il parametro "utente" ha un valore corrispondente ad uno di quelli consentiti, il parametro autore ha un valore corrispondente ad uno degli elementi presenti nella lista restituita da getListAutoriPerTipologia, il parametro tipologia ha un valore corrispondente ad uno degli elementi presenti nella lista restituita da getListTipologiaRisorse.	Deve essere restituito il valore booleano true.
CA01	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	getListTipologiaRisorse	Verificare che la lista delle tipologie sia correttamente prelevata e restituita sottoforma di stringhe.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql.	La funzione deve restituire una lista composta da esattamente quattro elementi, in ordine: - "Audio" - "Video" - "Immagine" - "Podcast"
CA02	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	getListAutori	Verificare che la lista delle tipologie sia correttamente prelevata e restituita sottoforma di stringhe.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql.	La funzione deve restituire una lista di esattamente 10 elementi, in ordine - "De Falco Occhio" - "Izu Kenro" - "Kubrick Stanley" - "Leone Sergio" - "McCurry Steve" - "Morricone Ennio" - "Mozart W. A." - "Scorzese Martin" - "van Beethoven Ludwig" - "Warhol Andy"
CA03	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	inserisciRisorsa	Verificare il comportamento della funzione quando essa viene chiamata prima che siano disponibili le informazioni relative alla tipologia tipologia	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql. La funzione viene chiamata prima che sia stata chiamata getListTipologiaRisorse.	Deve essere restituito il valore booleano false.
CA04	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	inserisciRisorsa	Verificare il comportamento della funzione quando essa viene chiamata prima che siano disponibili le informazioni relative agli autori	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql. La funzione viene chiamata prima che sia chiamata getListAutori.	Deve essere restituito il valore booleano false.

CA05	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	inserisciRisorsa	Verificare il comportamento della funzione quando ad essa vengono passati autore e tipologia diversi da quelli ottenuti precedentemente	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql. La funzione viene chiamata dopo getListTipologiaRisorse e getListAutori, ma usando valori diversi da quelli ottenuti mediante queste due funzioni.	Deve essere restituito il valore booleano false.
CA06	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	inserisciRisorsa	Verificare il comportamento della funzione quando ad essa viene passato uno o più parametri nulli	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql. Uno o più parametri della funzione sono posto a null.	Deve essere restituito il valore booleano false.
CA07	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	inserisciRisorsa	Verificare il comportamento della funzione quando essa viene chiamata con soli parametri corretti.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql. Alla funzione vengono passati parametri corretti.	Deve essere restituito il valore booleano true.
CA08	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	generaReport	Verificare il comportamento della funzione quando essa viene chiamata prima che siano disponibili le informazioni relative alla tipologia tipologia	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql. La funzione viene chiamata prima che sia stata chiamata getListTipologiaRisorse.	Deve essere restituito il valore zero
CA09	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	generaReport	Verificare il comportamento della funzione quando ad essa viene passato un valore per tipologia diverso da quello ottenuto precedentemente	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql. La funzione viene chiamata dopo getListTipologiaRisorse, ma usando valori diversi da quelli ottenuti mediante la chiamata a tale funzione.	Deve essere restituito il valore zero
CA10	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	generaReport	Verificare il comportamento della funzione quando ad essa viene passato uno o più parametri nulli	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql. Uno o più parametri della funzione sono posto a null.	Deve essere restituito il valore zero

CA11	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	generaReport	Verificare il comportamento della funzione quando essa viene chiamata con soli parametri corretti.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql. Alla funzione vengono passati parametri corretti. La funzione viene chiamata specificando il mese n°2 e la tipologia video.	Deve essere restituito 7.3.
CA12	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	getListaUtentiPremium	Verificare che la lista degli utenti premium sia correttamente generata.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql.	La lista restituita deve contenere un solo elemento: "Liguori Pietro".
CA13	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	getRisorseNonPossedute	Verificare il comportamento della funzione quando non sono disponibili informazioni circa gli utenti premium	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql. La funzione viene chiamata prima di getListaUtentiPremium.	La lista restituita deve essere vuota.
CA14	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	getRisorseNonPossedute	Verificare il comportamento della funzione quando uno o più parametri sono nulli	Il parametro utente della funzione è null.	Deve essere restituita una lista vuota
CA15	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	getRisorseNonPossedute	Verificare il comportamento della funzione quando uno o più dei parametri non sono corretti.	Il parametro utente della funzione ha un valore non appartenente a quelli ammessi.	Deve essere restituita una lista vuota
CA16	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	getRisorseNonPossedute	Verificare il comportamento della funzione quando uno o più dei parametri sono corretti. Verificare che la lista delle risorse non appartenenti ad un catalogo utente sia correttamente generata.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql. Alla funzione viene chiesto l'elenco delle risorse non appartenenti al catalogo dell'utente 'Barone Salvatore'	La funzione deve generare una lista di esattamente otto elementi, in ordine: -"The Ecstasy of Gold; Morricone Ennio" -"The Shining; Kubrick Stanley" -"The Aviator; Scorcese Martin" -"Cowboy solitari; Warhol Andy" -"Ragazza afgana; McCurry Steve" -"Requiem; Izu Kenro" -"Marilyn Monroe; Warhol Andy" -"The Hateful Eight Soundtrack; Morricone Ennio" di cui l'ultimo aggiunto al database al test n° 17.
CA17	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	aggiungiRisorseACatalogoUtente	Verificare il comportamento della funzione prima che siano disponibili le informazioni circa gli utenti premium.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql.	Deve essere restituito false.

CA18	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	aggiungiRisorseACatalogoUtente	Verificare il comportamento della funzione prima che siano disponibili le informazioni relative alle risorse non possedute da un utente.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql.	Deve essere restituito false.
CA19	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	aggiungiRisorseACatalogoUtente	verificare il comporamento della funzione quando gli vengono passati valori non corrispondenti a quelli ottenuti da database.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql.	Deve essere restituito false.
CA20	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	aggiungiRisorseACatalogoUtente	verificare il comporamento della funzione quando gli vengono passati riferimenti nulli.	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql.	Deve essere restituito false.
CA21	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	aggiungiRisorseACatalogoUtente	verificare il comportamento della funzione quando viene sollecitata da soli valori corretti	Il database è stato inizializzato con i comandi SQL contenuti nello script initDatabase.sql ed initPremium.sql.	Deve essere restituito true;
CA22	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	inserisciAutore	verificare il comporamento della funzione quando gli vengono passati riferimenti nulli.		Deve essere restituito false.
CA23	CoordinatoreAmministratoreTest	CoordinatoreAmministratore	inserisciAutore	verificare il comportamento della funzione quando viene sollecitata da soli valori corretti		Deve essere restituito true;

A Dati di test

idtipologia	descrizione
1	'Audio'
2	'Video'
3	'Immagine'
4	'Podcast'

(a) Tipologia

idautore	nome	cognome
5	'Ennio'	'Morricone'
6	'W. A.'	'Mozart'
7	'Ludwig'	'van Beethoven'
8	'Stanley'	'Kubrick'
9	'Martin'	'Scorzese'
10	'Sergio'	'Leone'
11	'Steve'	'McCurry'
12	'Kenro'	'Izu'
13	'Andy'	'Warhol'

(b) Autore

listaautori_idautore	tipologia_idtipologia
5	1
6	1
7	1
8	2
9	2
10	2
11	3
12	3
13	2
13	3

(c) Relazioni tra Tipologia ed Autore

idutente	nome	cognome	email	password	
24	'Salvatore'	'Barone'	'salvatorebarone@email.com'	'passw0rd'	27
25	'Alfonso'	'Di Martino'	'alfonsodimartino@email.com'	'Pa55uord'	28
26	'Pietro'	'Liguori'	'pilig@email.com'	'p4sswOrd'	29

(d) Utente

idcatalogo	idutente_fk
27	24
28	25
29	26

(e) CatalogoUtente

idpreferenza	idutente_fk	idautore_fk	idtipologia_fk
30	24	7	1
31	24	12	3
32	25	13	3
33	26	5	1
34	26	6	1

(f) PreferenzaUtente

idrisorsa	titolo	costoacquisto	costonoleggio	datapubblicazione
14	'The Ecstasy of Gold '	10.00	3.00	'2018-02-22'
15	'Eine Kleine Nachtmusik'	2.00	1.00	'2018-02-22'
16	'Symphony No. 9 '	2.00	1.00	'2018-02-22'
17	'The Shining'	12.00	4.50	'2018-02-22'
18	'The Aviator'	9	2	'2018-02-22'
19	'Il buono, il brutto,il cattivo'	7.00	5.5	'2018-02-22'
20	'Cowboy solitari'	4.5	0.3	'2018-02-22'
21	'Ragazza afgana'	3	1	'2018-02-22'
22	'Requiem'	5	0.7	'2018-02-22'
23	'Marilyn Monroe'	4.5	0.3	'2018-02-22'

(g) Risorsa

catalogoutente_idcatalogo	risorsepossedute_idrisorsa
27	23
27	18
28	15
28	16
28	22
29	16
29	15
29	19

(h) Relazioni tra CatalogoUtente e Risorsa

idacquisto	dataacquisto	idcatalogo_fk	idrisorsa_fk
36	'2018-02-22'	27	23
37	'2017-02-22'	27	18
38	'2018-02-22'	28	15
39	'2017-12-22'	28	16
40	'2017-12-22'	28	22
41	'2018-01-22'	29	16
42	'2018-01-22'	29	15
43	'2018-02-22'	29	19

(i) DateAcquisto

idnoleggio	datanoleggio	idcatalogo_fk	idrisorsa_fk
44	'2018-02-22'	27	21
45	'2017-02-22'	27	19
46	'2018-01-22'	28	14
47	'2018-02-22'	28	20
48	'2017-12-22'	28	23
49	'2017-12-22'	29	14
50	'2018-01-22'	29	21
51	'2018-02-22'	29	23

(j) DataNoleggio