Università degli Studi di Salerno **Corso di Ingegneria del Software**

***Test Case Specification DAO Document 1.0***

***23/12/2022***

****

0512107476 **– Salvatore Santoriello**0512107395 **– Bruno Farano**0512106126 **– Emanuele Milito**0512107080 **– Gianmichele Cancellaro**

***Revision History***

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autore** |
| 23/12/2022 | 1.0 | Prima stesura del documento | Membri del team |

***Indice***

**1.** Profilo

**2.** Ordine

**3.** Recensione

**4.** Prodotto

**5.** Carrello

**1.** Profilo

|  |  |
| --- | --- |
| *UserModelDS* | |
| *Metodo* | doSave(UserBean user) |
| *Test frame* | *US1* |
| *Parametri* | user |
| *Input* | user != null |
| *Output* | |
| Viene salvata all’interno del database un’istanza di UserBean, ossia l’oggetto restituito dalla procedura di registrazione, quindi un nuovo utente viene creato | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testdoSave() |
| *Output* | |
| Viene salvata all’interno del database un’istanza di UserBean, ossia l’oggetto restituito dalla procedura di registrazione, quindi un nuovo utente viene creato | |
| *Metodo* | doSave(UserBean user) |
| *Test frame* | *US2* |
| *Parametri* | user |
| *Input* | **null** |
| *Output* | |
| Non viene creata una nuova istanza di utente in quanto il parametro è null | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testdoSaveError() |
| *Output* | |
| Non viene creata una nuova istanza di utente in quanto il parametro è null | |
| *Metodo* | doRetrieveByKey(String email) |
| *Test frame* | *EM1* |
| *Parametri* | email |
| *Input* | admin@admin.it |
| *Output* | |
| Viene restituita un’istanza di UserBean associata al parametro email passato come parametro esplicito | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testdoRetrieveByKey() |
| *Output* | |
| Viene restituita un’istanza di UserBean associata al parametro email passato come parametro esplicito | |
| *Metodo* | doRetrieveByKey(String email) |
| *Test frame* | *EM2* |
| *Parametri* | email |
| *Input* | **pippopuppo@pippo.it** |
| *Output* | |
| Viene generato un errore in quanto il parametro email passato come parametro non è associato a nessun account all’interno del database | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testdoRetrieveByKeyError() |
| *Output* | |
| Viene generato un errore in quanto il parametro email passato come parametro non è associato a nessun account all’interno del database | |
| *Metodo* | doRetrieveByKey(String email) |
| *Test frame* | *EM3* |
| *Parametri* | email |
| *Input* | **null** |
| *Output* | |
| Viene generato un errore in quanto il parametro email è null | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testdoRetrieveByKeyNull() |
| *Output* | |
| Viene generato un errore in quanto il parametro email è vuoto o è null | |
| *Metodo* | doRetrieveAll(String order) |
| *Test frame* | OR1 |
| *Parametri* | order |
| *Input* | name |
| *Output* | |
| Viene restituita una lista di UserBean come risultato della query eseguita (ORDER BY order) in base al parametro order passato come parametro | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testdoRetrieveAll() |
| *Output* | |
| Viene restituita una lista di UserBean come risultato della query eseguita (ORDER BY order) in base al parametro order passato come parametro | |
| *Metodo* | doRetrieveAll(String order) |
| *Test frame* | OR2 |
| *Parametri* | order |
| *Input* | **school** |
| *Output* | |
| Viene restituita una lista UserBean vuota in quanto non ci sono account pertinenti alla query eseguita | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testdoRetrieveAllError() |
| *Output* | |
| Viene restituita una lista UserBean vuota in quanto non ci sono account pertinenti alla query eseguita | |
| *Metodo* | doRetrieveAll(String order) |
| *Test frame* | OR3 |
| *Parametri* | order |
| *Input* | **null** |
| *Output* | |
| Viene restituito null in quanto il parametro passato ha valore null | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testdoRetrieveAllNull() |
| *Output* | |
| Viene restituito null in quanto il parametro passato ha valore null | |
| *Metodo* | doUpdate(UserBean user) |
| *Test frame* | US1 |
| *Parametri* | user |
| *Input* | user != null |
| *Output* | |
| Viene aggiornata l’istanza all’interno del DB dell’utente associato al parametro user | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testdoUpdate() |
| *Output* | |
| Viene aggiornata l’istanza all’interno del DB dell’utente associato al parametro user | |
| *Metodo* | doUpdate(UserBean user) |
| *Test frame* | US2 |
| *Parametri* | user |
| *Input* | **user != null** |
| *Output* | |
| Viene generato un errore in quanto l’utente non è presente nel DB | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testdoUpdateError() |
| *Output* | |
| Viene generato un errore in quanto l’utente non è presente nel DB | |
| *Metodo* | doUpdate(UserBean user) |
| *Test frame* | US3 |
| *Parametri* | user |
| *Input* | **null** |
| *Output* | |
| Viene generato un errore in quanto il parametro user è null | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testdoUpdateNull() |
| *Output* | |
| Viene generato un errore in quanto il parametro user è null | |
| *Metodo* | doDelete(UserBean user) |
| *Test frame* | US1 |
| *Parametri* | user |
| *Input* | user != null |
| *Output* | |
| Viene eliminata l’istanza all’interno del DB dell’utente associato al parametro user | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testdoDelete() |
| *Output* | |
| Viene eliminata l’istanza all’interno del DB dell’utente associato al parametro user | |
| *Metodo* | doDelete(UserBean user) |
| *Test frame* | US2 |
| *Parametri* | user |
| *Input* | **user != null** |
| *Output* | |
| Viene generato un errore in quanto l’utente non è presente nel DB | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testdoDeleteError() |
| *Output* | |
| Viene generato un errore in quanto l’utente non è presente nel DB | |
| *Metodo* | doDelete(UserBean user) |
| *Test frame* | US3 |
| *Parametri* | user |
| *Input* | **null** |
| *Output* | |
| Viene generato un errore in quanto il parametro user è null | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testdoDeleteNull() |
| *Output* | |
| Viene generato un errore in quanto il parametro user è null | |
| *Metodo* | changeEmail(UserBean user, String newemail) |
| *Test frame* | US1-EM1 |
| *Parametri* | user, newemail |
| *Input* | user != null, ciaocomestai@live.it |
| *Output* | |
| Viene aggiornato il campo email dell’utente associato all’oggetto passato come parametro con il parametro newemail | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testchangeEmail() |
| *Output* | |
| Viene aggiornato il campo email dell’utente associato all’oggetto passato come parametro con il parametro newemail | |
| *Metodo* | changeEmail(UserBean user, String newemail) |
| *Test frame* | US2-EM1 |
| *Parametri* | user, newemail |
| *Input* | **user != null**,ciaocomestai@live.it |
| *Output* | |
| Viene generato un errore in quanto l’utente non è presente nel DB | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testchangeEmailError() |
| *Output* | |
| Viene generato un errore in quanto l’utente non è presente nel DB | |
| *Metodo* | changeEmail(UserBean user, String newemail) |
| *Test frame* | US3-EM1 |
| *Parametri* | user, newemail |
| *Input* | **null**,ciaocomestai@live.it |
| *Output* | |
| Viene generato un errore in quanto il parametro user è null | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testchangeEmailUserNull() |
| *Output* | |
| Viene generato un errore in quanto il parametro user è null | |
| *Metodo* | changeEmail(UserBean user, String newemail) |
| *Test frame* | US1-EM2 |
| *Parametri* | user, newemail |
| *Input* | admin@admin.it, **brunofarano@outlook.it** |
| *Output* | |
| Viene generato un errore in quanto l’email nuova è già presente nel DB | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testchangeEmailExists() |
| *Output* | |
| Viene generato un errore in quanto l’email nuova è già presente nel DB | |
| *Metodo* | changeName(UserBean user, String newname) |
| *Test frame* | US1-NN1 |
| *Parametri* | user, newname |
| *Input* | user != null, Carlo |
| *Output* | |
| Viene aggiornato il campo nome dell’utente associato all’oggetto passato come parametro con il parametro newname | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testchangeName() |
| *Output* | |
| Viene aggiornato il campo nome dell’utente associato all’oggetto passato come parametro con il parametro newname | |
| *Metodo* | changeName(UserBean user, String newname) |
| *Test frame* | US2-NN1 |
| *Parametri* | user, newname |
| *Input* | **user != null**, Carlo |
| *Output* | |
| Viene generato un errore in quanto l’utente non è presente nel DB | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testchangeNameError() |
| *Output* | |
| Viene generato un errore in quanto l’utente non è presente nel DB | |
| *Metodo* | changeName(UserBean user, String newname) |
| *Test frame* | US3-NN1 |
| *Parametri* | user, newname |
| *Input* | **null**, Carlo |
| *Output* | |
| Viene generato un errore in quanto il parametro user è null | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testchangeNameUserNull() |
| *Output* | |
| Viene generato un errore in quanto il parametro user è null | |
| *Metodo* | changeSurname(UserBean user, String newsurname) |
| *Test frame* | US1-NS1 |
| *Parametri* | user, newsurname |
| *Input* | user != null, Pellegatti |
| *Output* | |
| Viene aggiornato il campo cognome dell’utente associato all’oggetto passato come parametro con il parametro newsurname | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testchangeSurname() |
| *Output* | |
| Viene aggiornato il campo cognome dell’utente associato all’oggetto passato come parametro con il parametro newsurname | |
| *Metodo* | changeSurname(UserBean user, String newsurname) |
| *Test frame* | US2-NS1 |
| *Parametri* | user, newsurname |
| *Input* | **user != null**, Pellegatti |
| *Output* | |
| Viene generato un errore in quanto l’utente non è presente nel DB | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testchangeSurnameError() |
| *Output* | |
| Viene generato un errore in quanto l’utente non è presente nel DB | |
| *Metodo* | changeSurname(UserBean user, String newsurname) |
| *Test frame* | US3-NS1 |
| *Parametri* | user, newsurname |
| *Input* | **null**, Pellegatti |
| *Output* | |
| Viene generato un errore in quanto il parametro user è null | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testchangeSurnameUserNull() |
| *Output* | |
| Viene generato un errore in quanto il parametro user è null | |
| *Metodo* | changePassword(UserBean user, String newpassword) |
| *Test frame* | US1-NP1 |
| *Parametri* | user, newpassword |
| *Input* | user != null, Salvatore542! |
| *Output* | |
| Viene aggiornato il campo password dell’utente associato all’oggetto passato come parametro con il parametro newpassword | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testchangePassword() |
| *Output* | |
| Viene aggiornato il campo password dell’utente associato all’oggetto passato come parametro con il parametro newpassword | |
| *Metodo* | changePassword(UserBean user, String newpassword) |
| *Test frame* | US2-NP1 |
| *Parametri* | user, newpassword |
| *Input* | **user != null**, Salvatore542! |
| *Output* | |
| Viene generato un errore in quanto l’utente non è presente nel DB | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testchangePasswordError() |
| *Output* | |
| Viene generato un errore in quanto l’utente non è presente nel DB | |
| *Metodo* | changePassword(UserBean user, String newpassword) |
| *Test frame* | US3-NP1 |
| *Parametri* | user, newpassword |
| *Input* | **null**, Salvatore542! |
| *Output* | |
| Viene generato un errore in quanto il parametro user è null | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testchangePasswordUserNull() |
| *Output* | |
| Viene generato un errore in quanto il parametro user è null | |
| *Metodo* | changeAddress(UserBean user, String newaddress) |
| *Test frame* | US1-NA1 |
| *Parametri* | user, newaddress |
| *Input* | user != null, Via Principato, 52 |
| *Output* | |
| Viene aggiornato il campo indirizzo dell’utente associato all’oggetto passato come parametro con il parametro newaddress | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testchangeAddress() |
| *Output* | |
| Viene aggiornato il campo indirizzo dell’utente associato all’oggetto passato come parametro con il parametro newaddress | |
| *Metodo* | changeAddress(UserBean user, String newaddress) |
| *Test frame* | US2-NA1 |
| *Parametri* | user, newaddress |
| *Input* | **user != null**, Via Principato, 50 |
| *Output* | |
| Viene generato un errore in quanto l’utente non è presente nel DB | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testchangeAddressError() |
| *Output* | |
| Viene generato un errore in quanto l’utente non è presente nel DB | |
| *Metodo* | changeAddress(UserBean user, String newaddress) |
| *Test frame* | US3-NA1 |
| *Parametri* | user, newaddress |
| *Input* | **null**, Via Principato, 50 |
| *Output* | |
| Viene generato un errore in quanto il parametro user è null | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testchangeAddressUserNull() |
| *Output* | |
| Viene generato un errore in quanto il parametro user è null | |
| *Metodo* | changeCity(UserBean user, String newcity) |
| *Test frame* | US1-NC1 |
| *Parametri* | user, newcity |
| *Input* | user != null, Milano |
| *Output* | |
| Viene aggiornato il campo città dell’utente associato all’oggetto passato come parametro con il parametro newcity | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testchangeCity() |
| *Output* | |
| Viene aggiornato il campo città dell’utente associato all’oggetto passato come parametro con il parametro newcity | |
| *Metodo* | changeCity(UserBean user, String newcity) |
| *Test frame* | US2-NC1 |
| *Parametri* | user, newcity |
| *Input* | **user != null**, Milano |
| *Output* | |
| Viene generato un errore in quanto l’utente non è presente nel DB | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testchangeCityError() |
| *Output* | |
| Viene generato un errore in quanto l’utente non è presente nel DB | |
| *Metodo* | changeCity(UserBean user, String newcity) |
| *Test frame* | US3-NC1 |
| *Parametri* | user, newcity |
| *Input* | **null**, Milano |
| *Output* | |
| Viene generato un errore in quanto il parametro user è null | |
| *Test driver* | |
| *Classe* | UserModelDSTest |
| *Metodo* | testchangeCityUserNull() |
| *Output* | |
| Viene generato un errore in quanto il parametro user è null | |

|  |  |
| --- | --- |
| *LoginModelDS* | |
| *Metodo* | doRetrieveByKey(String email, String password) |
| *Test frame* | *EM1-P1* |
| *Parametri* | email, password |
| *Input* | admin@admin.it, admin |
| *Output* | |
| Viene restituito l’oggetto UserBean che ha come parametri email e password quelli passati come parametro | |
| *Test driver* | |
| *Classe* | LoginModelDSTest |
| *Metodo* | testdoRetrieveByKey() |
| *Output* | |
| Viene salvata all’interno del database un’istanza di UserBean, ossia l’oggetto restituito dalla procedura di registrazione, quindi un nuovo utente viene creato | |
| *Metodo* | doRetrieveByKey(String email, String password) |
| *Test frame* | *EM2-P1* |
| *Parametri* | email, password |
| *Input* | **pippopuppo@pippo.it**, Salvatore610! |
| *Output* | |
| Viene generato un errore in quanto l’email non è presente nel DB | |
| *Test driver* | |
| *Classe* | LoginModelDSTest |
| *Metodo* | testdoRetrieveByKeyError() |
| *Output* | |
| Viene generato un errore in quanto l’email non è presente nel DB | |
| *Metodo* | doRetrieveByKey(String email, String password) |
| *Test frame* | *EM3-P1* |
| *Parametri* | email, password |
| *Input* | **null**, Salvatore610! |
| *Output* | |
| Viene generato un errore in quanto il parametro email è null | |
| *Test driver* | |
| *Classe* | LoginModelDSTest |
| *Metodo* | testdoRetrieveByKeyEmailNull() |
| *Output* | |
| Viene generato un errore in quanto il parametro email è null | |
| *Metodo* | doRetrieveByKey(String email, String password) |
| *Test frame* | *EM1-P2* |
| *Parametri* | email, password |
| *Input* | admin@admin,it, **Pippo** |
| *Output* | |
| Viene generato un errore in quanto la password non è associata alla email passata come parametro (password errata) | |
| *Test driver* | |
| *Classe* | LoginModelDSTest |
| *Metodo* | testdoRetrieveByKeyPasswordError() |
| *Output* | |
| Viene generato un errore in quanto la password non è associata alla email passata come parametro (password errata) | |
| *Metodo* | doRetrieveByKey(String email, String password) |
| *Test frame* | *EM1-P3* |
| *Parametri* | email, password |
| *Input* | admin@admin,it, **null** |
| *Output* | |
| Viene generato un errore in quanto il parametro password è null | |
| *Test driver* | |
| *Classe* | LoginModelDSTest |
| *Metodo* | testdoRetrieveByKeyPasswordNull() |
| *Output* | |
| Viene generato un errore in quanto il parametro password è null | |

**2.** Ordine

|  |  |
| --- | --- |
| *OrdiniModelDS* | |
| *Metodo* | doSave(OrdiniBean ordine) |
| *Test frame* | *OR1* |
| *Parametri* | ordine |
| *Input* | ordine != null |
| *Output* | |
| Viene salvata all’interno del database un’istanza di OrdiniBean, ossia l’oggetto restituito dalla procedura di acquisto, quindi un nuovo ordine viene creato e associato all’utente che l’ha creato | |
| *Test driver* | |
| *Classe* | OrdiniModelDSTest |
| *Metodo* | testdoSave() |
| *Output* | |
| Viene salvata all’interno del database un’istanza di OrdiniBean, ossia l’oggetto restituito dalla procedura di acquisto, quindi un nuovo ordine viene creato e associato all’utente che l’ha creato | |
| *Metodo* | doSave(OrdiniBean ordine) |
| *Test frame* | OR2 |
| *Parametri* | ordine |
| *Input* | **null** |
| *Output* | |
| Non viene creata una nuova istanza di utente in quanto il parametro è null | |
| *Test driver* | |
| *Classe* | OrdiniModelDSTest |
| *Metodo* | testdoSaveNull() |
| *Output* | |
| Non viene creata una nuova istanza di utente in quanto il parametro è null | |
| *Metodo* | doDelete(int id) |
| *Test frame* | *ID1* |
| *Parametri* | id |
| *Input* | 1 |
| *Output* | |
| Viene eliminato l’ordine che ha come parametro ID quello passato come parametro esplicito | |
| *Test driver* | |
| *Classe* | OrdiniModelDSTest |
| *Metodo* | testdoDelete() |
| *Output* | |
| Viene eliminato l’ordine che ha come parametro ID quello passato come parametro esplicito | |
| *Metodo* | doDelete(int id) |
| *Test frame* | ID2 |
| *Parametri* | id |
| *Input* | **434** |
| *Output* | |
| Viene generato un errore in quanto non ci sono corrispondenze all’interno del database dell’id passato come parametro | |
| *Test driver* | |
| *Classe* | OrdiniModelDSTest |
| *Metodo* | testdoDeleteError() |
| *Output* | |
| Viene generato un errore in quanto non ci sono corrispondenze all’interno del database dell’id passato come parametro | |
| *Metodo* | doRetrieveAll(String email) |
| *Test frame* | *EM1* |
| *Parametri* | email |
| *Input* | brunofarano@outlook.it |
| *Output* | |
| Viene restituita una lista di OrdiniBean associata all’utente che ha come email quella passata come parametro | |
| *Test driver* | |
| *Classe* | OrdiniModelDSTest |
| *Metodo* | testdoRetrieveAll() |
| *Output* | |
| Viene restituita una lista di OrdiniBean associata all’utente che ha come email quella passata come parametro | |
| *Metodo* | doRetrieveAll(String email) |
| *Test frame* | *EM2* |
| *Parametri* | email |
| *Input* | **pippopuppo@pippo.it** |
| *Output* | |
| Viene restituita una lista vuota in quanto non ci sono utenti che hanno come email quella passata come parametro | |
| *Test driver* | |
| *Classe* | OrdiniModelDSTest |
| *Metodo* | testdoRetrieveAllError() |
| *Output* | |
| Viene restituita una lista vuota in quanto non ci sono utenti che hanno come email quella passata come parametro | |
| *Metodo* | doUpdateDelivery(int id, String spedizione) |
| *Test frame* | ID1-SP1 |
| *Parametri* | id, spedizione |
| *Input* | 1, Spedizione arrivata con successo! |
| *Output* | |
| Viene aggiornata la spedizione dell’ordine che ha come id quello passato come parametro | |
| *Test driver* | |
| *Classe* | OrdiniModelDSTest |
| *Metodo* | testdoUpdateDelivery() |
| *Output* | |
| Viene aggiornata la spedizione dell’ordine che ha come id quello passato come parametro | |
| *Metodo* | doUpdateDelivery(int id, String spedizione) |
| *Test frame* | ID2-SP1 |
| *Parametri* | id, spedizione |
| *Input* | **3434**, Spedizione arrivata con successo! |
| *Output* | |
| Viene generato un errore in quanto non ci sono ordini che hanno come corrispondenza l’id passato come parametro | |
| *Test driver* | |
| *Classe* | OrdiniModelDSTest |
| *Metodo* | testdoUpdateDeliveryError() |
| *Output* | |
| Viene generato un errore in quanto non ci sono ordini che hanno come corrispondenza l’id passato come parametro | |
| *Metodo* | doUpdateOrder(int id, String ordine) |
| *Test frame* | ID1-OR1 |
| *Parametri* | id, ordine |
| *Input* | 1, Spedizione ritirata dal corriere |
| *Output* | |
| Viene aggiornata la descrizione dell’ordine che ha come id quello passato come parametro | |
| *Test driver* | |
| *Classe* | OrdiniModelDSTest |
| *Metodo* | testdoUpdateOrder() |
| *Output* | |
| Viene aggiornata la descrizione dell’ordine che ha come id quello passato come parametro | |
| *Metodo* | doUpdateOrder(int id, String ordine) |
| *Test frame* | ID2-OR1 |
| *Parametri* | id, ordine |
| *Input* | **434**, Spedizione ritirata dal corriere |
| *Output* | |
| Viene generato un errore in quanto l’ordine che ha come id quello passato come parametro non è presente nel DB | |
| *Test driver* | |
| *Classe* | OrdiniModelDSTest |
| *Metodo* | testdoUpdateOrderError() |
| *Output* | |
| Viene generato un errore in quanto l’ordine che ha come id quello passato come parametro non è presente nel DB | |

|  |  |
| --- | --- |
| *OrdineDettaglioModelDS* | |
| *Metodo* | doSave(OrdineDettaglioBean prodotto) |
| *Test frame* | P1 |
| *Parametri* | prodotto |
| *Input* | prodotto != null |
| *Output* | |
| Viene salvata all’interno del database un’istanza di OrdineDettaglioBean, ossia un oggetto che rappresenta un prodotto acquistato da un utente, associato all’ordine principale tramite id e all’utente tramite email | |
| *Test driver* | |
| *Classe* | OrdineDettaglioModelDSTest |
| *Metodo* | testdoSave() |
| *Output* | |
| Viene salvata all’interno del database un’istanza di OrdineDettaglioBean, ossia un oggetto che rappresenta un prodotto acquistato da un utente, associato all’ordine principale tramite id e all’utente tramite email | |
| *Metodo* | doSave(OrdineDettaglioBean prodotto) |
| *Test frame* | P2 |
| *Parametri* | prodotto |
| *Input* | **null** |
| *Output* | |
| Viene generato un errore in quanto il parametro prodotto è null | |
| *Test driver* | |
| *Classe* | OrdineDettaglioModelDSTest |
| *Metodo* | testdoSaveNull() |
| *Output* | |
| Viene generato un errore in quanto il parametro prodotto è null | |
| *Metodo* | doRetrieveByKey(int id) |
| *Test frame* | *ID1* |
| *Parametri* | Id |
| *Input* | 1 |
| *Output* | |
| Viene restituita la lista dei prodotti associati all’ordine che ha come id quello passato come parametro | |
| *Test driver* | |
| *Classe* | OrdineDettaglioModelDSTest |
| *Metodo* | testdoRetrieveByKey() |
| *Output* | |
| Viene restituita la lista dei prodotti associati all’ordine che ha come id quello passato come parametro | |
| *Metodo* | doRetrieveByKey(int id) |
| *Test frame* | *ID2* |
| *Parametri* | Id |
| *Input* | **232** |
| *Output* | |
| Viene generato un errore in quanto non ci sono corrispondenze all’interno del database dell’id passato come parametro | |
| *Test driver* | |
| *Classe* | OrdineDettaglioModelDSTest |
| *Metodo* | testdoRetrieveByKeyError() |
| *Output* | |
| Viene generato un errore in quanto non ci sono corrispondenze all’interno del database dell’id passato come parametro | |

**3.** Recensione

|  |  |
| --- | --- |
| *RecensioniModelDS* | |
| *Metodo* | doSave(RecensioniBean recensione) |
| *Test frame* | R1 |
| *Parametri* | recensione |
| *Input* | recensione != null |
| *Output* | |
| Viene salvata all’interno del database un’istanza di RecensioneBean, ossia una nuova recensione e associata all’account che l’ha creata | |
| *Test driver* | |
| *Classe* | RecensioniModelDSTest |
| *Metodo* | testdoSave() |
| *Output* | |
| Viene salvata all’interno del database un’istanza di RecensioneBean, ossia una nuova recensione e associata all’account che l’ha creata | |
| *Metodo* | doSave(RecensioniBean recensione) |
| *Test frame* | R2 |
| *Parametri* | recensione |
| *Input* | **null** |
| *Output* | |
| Non viene creata una nuova istanza di recensione in quanto il parametro è null | |
| *Test driver* | |
| *Classe* | RecensioniModelDSTest |
| *Metodo* | testdoSaveNull() |
| *Output* | |
| Non viene creata una nuova istanza di recensione in quanto il parametro è null | |
| *Metodo* | doDelete(int numero) |
| *Test frame* | N1 |
| *Parametri* | numero |
| *Input* | 1 |
| *Output* | |
| Viene eliminato la recensione che ha come numero quello passato come parametro esplicito | |
| *Test driver* | |
| *Classe* | RecensioniModelDSTest |
| *Metodo* | testdoDelete() |
| *Output* | |
| Viene eliminato la recensione che ha come numero quello passato come parametro esplicito | |
| *Metodo* | doDelete(int numero) |
| *Test frame* | N2 |
| *Parametri* | numero |
| *Input* | **34342** |
| *Output* | |
| Viene generato un errore in quanto non ci sono corrispondenze all’interno del database del numero passato come parametro | |
| *Test driver* | |
| *Classe* | RecensioniModelDSTest |
| *Metodo* | testdoDeleteError() |
| *Output* | |
| Viene generato un errore in quanto non ci sono corrispondenze all’interno del database del numero passato come parametro | |
| *Metodo* | doRetrieveByKey(String nomeprodotto) |
| *Test frame* | NP1 |
| *Parametri* | nomeprodotto |
| *Input* | Maglia Milan Home 2022/2023 |
| *Output* | |
| Viene restituita la recensione associata al prodotto che ha come nome quello passato come parametro | |
| *Test driver* | |
| *Classe* | RecensioniModelDSTest |
| *Metodo* | testdoRetrieveByKey() |
| *Output* | |
| Viene restituita la recensione associata al prodotto che ha come nome quello passato come parametro | |
| *Metodo* | doRetrieveByKey(String nomeprodotto) |
| *Test frame* | NP2 |
| *Parametri* | nomeprodotto |
| *Input* | **Pippo e Paperino 02323** |
| *Output* | |
| Viene restituito null in quanto non ci sono prodotti che hanno come nome quello passato come parametro | |
| *Test driver* | |
| *Classe* | RecensioniModelDSTest |
| *Metodo* | testdoRetrieveByKeyError() |
| *Output* | |
| Viene restituito null in quanto non ci sono prodotti che hanno come nome quello passato come parametro | |
| *Metodo* | doRetrieveAll(String nome) |
| *Test frame* | N1 |
| *Parametri* | nome |
| *Input* | Maglia Milan Home 2022/2023 |
| *Output* | |
| Viene restituita una lista di RecensioniBean associata al prodotto che ha come nome quello passato come parametro | |
| *Test driver* | |
| *Classe* | RecensioniModelDSTest |
| *Metodo* | testdoRetrieveAll() |
| *Output* | |
| Viene restituita una lista di RecensioniBean associata al prodotto che ha come nome quello passato come parametro | |
| *Metodo* | doRetrieveAll(String nome) |
| *Test frame* | N2 |
| *Parametri* | nome |
| *Input* | **Pippo e Paperino 230203** |
| *Output* | |
| Viene restituita una lista vuota in quanto non ci sono corrispondenze all’interno del DB di prodotti che hanno il nome quello passato come parametro | |
| *Test driver* | |
| *Classe* | RecensioniModelDSTest |
| *Metodo* | testdoRetrieveAllError() |
| *Output* | |
| Viene restituita una lista vuota in quanto non ci sono corrispondenze all’interno del DB di prodotti che hanno il nome quello passato come parametro | |

**4.** Prodotto

|  |  |
| --- | --- |
| *ProdottoModelDS* | |
| *Metodo* | doSave(ProdottoBean prodotto, String email) |
| *Test frame* | P1-E1 |
| *Parametri* | prodotto, email |
| *Input* | prodotto != null, admin@admin.it |
| *Output* | |
| Viene aggiunto una nuova istanza di ProdottoBean all’interno del database, ossia viene creato un nuovo prodotto da mettere in vendita da parte dell’admin | |
| *Test driver* | |
| *Classe* | ProdottoModelDSTest |
| *Metodo* | testdoSave() |
| *Output* | |
| Viene aggiunto una nuova istanza di ProdottoBean all’interno del database, ossia viene creato un nuovo prodotto da mettere in vendita da parte dell’admin | |
| *Metodo* | doSave(ProdottoBean prodotto, String email) |
| *Test frame* | P2-E1 |
| *Parametri* | prodotto, email |
| *Input* | **null**, admin@admin.it |
| *Output* | |
| Non viene creata una nuova istanza di prodotto in quanto il parametro prodotto è null | |
| *Test driver* | |
| *Classe* | ProdottoModelDSTest |
| *Metodo* | testdoSaveProductNull() |
| *Output* | |
| Non viene creata una nuova istanza di prodotto in quanto il parametro prodotto è null | |
| *Metodo* | doRetrieveByKey(int code) |
| *Test frame* | C1 |
| *Parametri* | code |
| *Input* | 1 |
| *Output* | |
| Viene restituita un’istanza di ProdottoBean del prodotto che ha come codice quello passato come parametro | |
| *Test driver* | |
| *Classe* | ProdottoModelDSTest |
| *Metodo* | testdoRetrieveByKey() |
| *Output* | |
| Viene restituita un’istanza di ProdottoBean del prodotto che ha come codice quello passato come parametro | |
| *Metodo* | doRetrieveByKey(int code) |
| *Test frame* | C2 |
| *Parametri* | code |
| *Input* | **2323** |
| *Output* | |
| Viene generato un errore in quanto il parametro codice passato come parametro non è associato a nessun prodotto all’interno del database | |
| *Test driver* | |
| *Classe* | ProdottoModelDSTest |
| *Metodo* | testdoRetrieveByKeyError() |
| *Output* | |
| Viene generato un errore in quanto il parametro codice passato come parametro non è associato a nessun prodotto all’interno del database | |
| *Metodo* | doDelete(int code, String email) |
| *Test frame* | C1-EM1 |
| *Parametri* | code, email |
| *Input* | 1, admin@admin.it |
| *Output* | |
| Viene eliminato il prodotto dal database che ha come codice quello passato come parametro | |
| *Test driver* | |
| *Classe* | ProdottoModelDSTest |
| *Metodo* | testdoDelete() |
| *Output* | |
| Viene eliminato il prodotto dal database che ha come codice quello passato come parametro | |
| *Metodo* | doDelete(int code, String email) |
| *Test frame* | C2-EM1 |
| *Parametri* | code, email |
| *Input* | **2323**, admin@admin.it |
| *Output* | |
| Viene generato un errore in quanto non è presente nessun prodotto ha come codice quello passato come parametro | |
| *Test driver* | |
| *Classe* | ProdottoModelDSTest |
| *Metodo* | testdoDeleteError() |
| *Output* | |
| Viene generato un errore in quanto non è presente nessun prodotto ha come codice quello passato come parametro | |
| *Metodo* | doRetrieveAll(String ordine) |
| *Test frame* | OR1 |
| *Parametri* | ordine |
| *Input* | nome |
| *Output* | |
| Viene restituita una lista di ProdottoBean contenente i prodotti che sono pertinenti (ORDER BY order) al parametro order | |
| *Test driver* | |
| *Classe* | ProdottoModelDSTest |
| *Metodo* | testdoRetrieveAll() |
| *Output* | |
| Viene restituita una lista di ProdottoBean contenente i prodotti che sono pertinenti (ORDER BY order) al parametro order | |
| *Metodo* | doRetrieveAll(String ordine) |
| *Test frame* | OR2 |
| *Parametri* | ordine |
| *Input* | **famiglia** |
| *Output* | |
| Viene restituita una lista vuota in quanto il parametro ordine non è pertinente | |
| *Test driver* | |
| *Classe* | ProdottoModelDSTest |
| *Metodo* | testdoRetrieveAllError() |
| *Output* | |
| Viene restituita una lista vuota in quanto il parametro ordine non è pertinente | |
| *Metodo* | doUpdatePrice(int code, int prezzo) |
| *Test frame* | C1-P1 |
| *Parametri* | code, prezzo |
| *Input* | 1, 230 |
| *Output* | |
| Viene aggiornato il prezzo con il parametro prezzo passato come parametro del prodotto che ha come codice quello passato come parametro | |
| *Test driver* | |
| *Classe* | ProdottoModelDSTest |
| *Metodo* | testdoUpdatePrice() |
| *Output* | |
| Viene aggiornato il prezzo con il parametro prezzo passato come parametro del prodotto che ha come codice quello passato come parametro | |
| *Metodo* | doUpdatePrice(int code, int prezzo) |
| *Test frame* | C2-P1 |
| *Parametri* | code, prezzo |
| *Input* | **232**, 2320 |
| *Output* | |
| Viene generato un errore in quanto il prodotto non è presente nel DB | |
| *Test driver* | |
| *Classe* | ProdottoModelDSTest |
| *Metodo* | testdoUpdatePriceError() |
| *Output* | |
| Viene generato un errore in quanto il prodotto non è presente nel DB | |
| *Metodo* | doRetrieveSearch(String ricerca) |
| *Test frame* | R1 |
| *Parametri* | ricerca |
| *Input* | Maglia |
| *Output* | |
| Viene restituita una lista di ProdottoBean in base alla ricerca passata come parametro, ossia una lista dei prodotti pertinenti alla ricerca | |
| *Test driver* | |
| *Classe* | ProdottoModelDSTest |
| *Metodo* | testdoRetrieveSearch() |
| *Output* | |
| Viene restituita una lista di ProdottoBean in base alla ricerca passata come parametro, ossia una lista dei prodotti pertinenti alla ricerca | |
| *Metodo* | doRetrieveSearch(String ricerca) |
| *Test frame* | R2 |
| *Parametri* | ricerca |
| *Input* | **Basket** |
| *Output* | |
| Viene generato un errore in quanto non ci sono prodotti pertinenti alla ricerca | |
| *Test driver* | |
| *Classe* | ProdottoModelDSTest |
| *Metodo* | testdoRetrieveSearchError() |
| *Output* | |
| Viene generato un errore in quanto non ci sono prodotti pertinenti alla ricerca | |
| *Metodo* | doRetrieveAllProducts() |
| *Test frame* | // |
| *Parametri* | // |
| *Input* | // |
| *Output* | |
| Viene restituita la lista di tutti i prodotti presenti all’interno del database | |
| *Test driver* | |
| *Classe* | ProdottoModelDSTest |
| *Metodo* | testdoRetrieveAllProducts() |
| *Output* | |
| Viene restituita la lista di tutti i prodotti presenti all’interno del database | |
| *Metodo* | doUpdateQuantity(int code, int quantità) |
| *Test frame* | C1-Q1 |
| *Parametri* | code, quantità |
| *Input* | 1, 23 |
| *Output* | |
| Viene aggiornato il campo quantità del prodotto associato al codice passato come parametro con il parametro quantità | |
| *Test driver* | |
| *Classe* | ProdottoModelDSTest |
| *Metodo* | testdoUpdateQuantity() |
| *Output* | |
| Viene aggiornato il campo quantità del prodotto associato al codice passato come parametro con il parametro quantità | |
| *Metodo* | doUpdateQuantity(int code, int quantità) |
| *Test frame* | C2-Q1 |
| *Parametri* | code, quantità |
| *Input* | **2323**, 23 |
| *Output* | |
| Viene generato un errore in quanto il prodotto non è presente nel DB | |
| *Test driver* | |
| *Classe* | ProdottoModelDSTest |
| *Metodo* | testdoUpdateQuantityError() |
| *Output* | |
| Viene generato un errore in quanto il prodotto non è presente nel DB | |
| *Metodo* | doIncreaseQuantity(int code, int quantità) |
| *Test frame* | C1-Q1 |
| *Parametri* | code, quantità |
| *Input* | 1, 231 |
| *Output* | |
| Viene aggiornato il campo quantità del prodotto associato al codice passato come parametro con il parametro quantità | |
| *Test driver* | |
| *Classe* | ProdottoModelDSTest |
| *Metodo* | testdoIncreaseQuantity() |
| *Output* | |
| Viene aggiornato il campo quantità del prodotto associato al codice passato come parametro con il parametro quantità | |
| *Metodo* | doIncreaseQuantity(int code, int quantità) |
| *Test frame* | C2-Q1 |
| *Parametri* | code, quantità |
| *Input* | **2323**, 231 |
| *Output* | |
| Viene generato un errore in quanto il prodotto non è presente nel DB | |
| *Test driver* | |
| *Classe* | ProdottoModelDSTest |
| *Metodo* | testdoIncreaseQuantityError() |
| *Output* | |
| Viene generato un errore in quanto il prodotto non è presente nel DB | |

**5.** Carrello

Rappresenta l’entità Carrello, questa, essendo volatile poiché istanziata solo quando l’utente effettua la procedura di login e assegnata a quest’ultimo mediante sessione, non rispetta il modello DAO poiché non c’è bisogno di persistere i dati relativi al carrello.