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

***Test Plan DAO Document 1.0***

***19/12/2022***

****

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

***Revision History***

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autore** |
| 19/12/2022 | 1.0 | Prima stesura del documento | Membri del team |

***Indice***

**1.** Test case DAO

**1.1***Profilo* **1.2***Ordine*  
 **1.3***Recensione* **1.4***Prodotto* **1.5***Carrello*

**1.** Test case DAO

**1.1***Profilo*

|  |  |
| --- | --- |
| *UserModelDS* | |
| *Metodo* | doSave(UserBean user) |
| *Parametri* | user |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *user* | US1: user != null [property user\_ok]  US2: user == null [property user\_error] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *US1* | Viene salvata all’interno del database un’istanza di UserBean, ossia l’oggetto restituito dalla procedura di registrazione, quindi un nuovo utente viene creato |
| *US2* | Non viene creata una nuova istanza di utente in quanto il parametro è null |
| *Metodo* | doRetrieveByKey(String email) |
| *Parametri* | email |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *email* | EM1: email presente nel DB e associata ad un account [property email\_ok]  EM2: email non presente nel DB [property email\_error]  EM3: email == null || email == “” [property email\_empty] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *EM1* | Viene restituita un’istanza di UserBean associata al parametro email passato come parametro esplicito |
| *EM2* | Viene generato un errore in quanto il parametro email passato come parametro non è associato a nessun account all’interno del database |
| *EM3* | Viene generato un errore in quanto il parametro email è vuoto o è null |
| *Metodo* | doRetrieveAll(String order) |
| *Parametri* | order |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *order* | OR1: order != null e pertinente ad account nel DB [property order\_exists]  OR2 = order != null ma non pertinente ad account nel DB [property order\_no]  OR3: order == null [property order\_null] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *OR1* | Viene restituita una lista di UserBean come risultato della query eseguita (ORDER BY order) in base al parametro order passato come parametro |
| *OR2* | Viene restituita una lista UserBean vuota in quanto non ci sono account pertinenti alla query eseguita |
| *OR3* | Viene restituito null in quanto il parametro passato ha valore null |
| *Metodo* | doUpdate(UserBean user) |
| *Parametri* | user |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *user* | US1: user != null e associato a utente nel DB [property user\_ok]  US2: user != null ma non associato ad utente [property user\_notexists]  US3: user == null |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *US1* | Viene aggiornata l’istanza all’interno del DB dell’utente associato al parametro user |
| *US2* | Viene generato un errore in quanto l’utente non è presente nel DB |
| *US3* | Viene generato un errore in quanto il parametro user è null |
| *Metodo* | doDelete(UserBean user) |
| *Parametri* | user |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *user* | US1: user != null e associato a utente nel DB [property user\_ok]  US2: user != null ma non associato ad utente [property user\_notexists]  US3: user == null |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *US1* | Viene eliminata l’istanza all’interno del DB dell’utente associato al parametro user |
| *US2* | Viene generato un errore in quanto l’utente non è presente nel DB |
| *US3* | Viene generato un errore in quanto il parametro user è null |
| *Metodo* | changeEmail(UserBean user, String newemail) |
| *Parametri* | user, newemail |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *user* | US1: user != null e associato a utente nel DB [property user\_ok]  US2: user != null ma non associato ad utente [property user\_notexists]  US3: user == null |
| *newemail* | EM1: email non presente nel DB [property email\_ok]  EM2: email presente nel DB [property email\_error] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *US1-EM1* | Viene aggiornato il campo email dell’utente associato all’oggetto passato come parametro con il parametro newemail |
| *US2-EM1* | Viene generato un errore in quanto l’utente non è presente nel DB |
| *US3-EM1* | Viene generato un errore in quanto il parametro user è null |
| *US1-EM2* | Viene generato un errore in quanto l’email nuova è già presente nel DB |
| *Metodo* | changeName(UserBean user, String newname) |
| *Parametri* | user, newname |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *user* | US1: user != null e associato a utente nel DB [property user\_ok]  US2: user != null ma non associato ad utente [property user\_notexists]  US3: user == null |
| *newname* | NN1: newname != null e formato rispettato [property name\_ok] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *US1-NN1* | Viene aggiornato il campo nome dell’utente associato all’oggetto passato come parametro con il parametro newname |
| *US2-NN1* | Viene generato un errore in quanto l’utente non è presente nel DB |
| *US3-NN1* | Viene generato un errore in quanto il parametro user è null |
| *Metodo* | changeSurname(UserBean user, String newsurname) |
| *Parametri* | user, newsurname |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *user* | US1: user != null e associato a utente nel DB [property user\_ok]  US2: user != null ma non associato ad utente [property user\_notexists]  US3: user == null |
| *newsurname* | NS1: newsurname != null e formato rispettato [property surname\_ok] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *US1-NS1* | Viene aggiornato il campo cognome dell’utente associato all’oggetto passato come parametro con il parametro newsurname |
| *US2-NS1* | Viene generato un errore in quanto l’utente non è presente nel DB |
| *US3-NS1* | Viene generato un errore in quanto il parametro user è null |
| *Metodo* | changePassword(UserBean user, String newpassword) |
| *Parametri* | user, newpassword |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *user* | US1: user != null e associato a utente nel DB [property user\_ok]  US2: user != null ma non associato ad utente [property user\_notexists]  US3: user == null |
| *newpassword* | NP1: newpassword != null e formato rispettato [property password\_ok] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *US1-NP1* | Viene aggiornato il campo password dell’utente associato all’oggetto passato come parametro con il parametro newpassword |
| *US2-NP1* | Viene generato un errore in quanto l’utente non è presente nel DB |
| *US3-NP1* | Viene generato un errore in quanto il parametro user è null |
| *Metodo* | changeAddress(UserBean user, String newaddress) |
| *Parametri* | user, newaddress |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *user* | US1: user != null e associato a utente nel DB [property user\_ok]  US2: user != null ma non associato ad utente [property user\_notexists]  US3: user == null |
| *newaddress* | NA1: newaddress != null e formato rispettato [property address\_ok] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *US1-NA1* | Viene aggiornato il campo indirizzo dell’utente associato all’oggetto passato come parametro con il parametro newaddress |
| *US2-NA1* | Viene generato un errore in quanto l’utente non è presente nel DB |
| *US3-NA1* | Viene generato un errore in quanto il parametro user è null |
| *Metodo* | changeCity(UserBean user, String newcity) |
| *Parametri* | user, newcity |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *user* | US1: user != null e associato a utente nel DB [property user\_ok]  US2: user != null ma non associato ad utente [property user\_notexists]  US3: user == null |
| *newcity* | NC1: newcity != null e formato rispettato [property city\_ok] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *US1-NC1* | Viene aggiornato il campo città dell’utente associato all’oggetto passato come parametro con il parametro newcity |
| *US2-NC1* | Viene generato un errore in quanto l’utente non è presente nel DB |
| *US3-NC1* | Viene generato un errore in quanto il parametro user è null |

|  |  |
| --- | --- |
| *LoginModelDS* | |
| *Metodo* | doRetrieveByKey(String email, String password) |
| *Parametri* | email, password |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *email* | EM1: email presente nel DB e associata ad un account [property email\_ok]  EM2: email non presente nel DB [property email\_error]  EM3: email == null || email == “” [property email\_empty] |
| *password* | P1: password != null e associata ad email [property password\_ok]  P2: password != null ma non associata ad email [property password\_wrong]  P3: password== null || password == “” [property password\_empty] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *EM1-P1* | Viene restituito l’oggetto UserBean che ha come parametri email e password quelli passati come parametro |
| *EM2-P1* | Viene generato un errore in quanto l’email non è presente nel DB |
| *EM3-P1* | Viene generato un errore in quanto il parametro email è null |
| *EM1-P2* | Viene generato un errore in quanto la password non è associata alla email passata come parametro (password errata) |
| *EM1-P3* | Viene generato un errore in quanto il parametro password è null |

**1.2** *Ordine*

|  |  |
| --- | --- |
| *OrdiniModelDS* | |
| *Metodo* | doSave(OrdiniBean ordine) |
| *Parametri* | ordine |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *ordine* | OR1: ordine != null [property ordine\_ok]  OR2: ordine == null [property ordine\_error] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *OR1* | 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 |
| *OR2* | Non viene creata una nuova istanza di utente in quanto il parametro è null |
| *Metodo* | doDelete(int id) |
| *Parametri* | id |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *id* | ID1: id presente nel DB e associata ad un ordine [property id\_ok]  ID2: id non presente nel DB [property id\_error] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *ID1* | Viene eliminato l’ordine che ha come parametro ID quello passato come parametro esplicito |
| *ID2* | Viene generato un errore in quanto non ci sono corrispondenze all’interno del database dell’id passato come parametro |
| *Metodo* | doRetrieveAll(String email) |
| *Parametri* | email |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *email* | EM1: email != null e associata ad un account nel database [property email\_ok]  EM2: email != null ma non associata a nessun account [property email\_notexists] EM3: email == null || email == “” [property email\_error] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *EM1* | Viene restituita una lista di OrdiniBean associata all’utente che ha come email quella passata come parametro |
| *EM2* | 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) |
| *Parametri* | id, spedizione |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *id* | ID1: id presente nel DB e associata ad un ordine [property id\_ok]  ID2: id non presente nel DB [property id\_error] |
| *spedizione* | SP1: spedizione != null [property delivery\_ok] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *ID1-SP1* | Viene aggiornata la spedizione dell’ordine che ha come id quello passato come parametro |
| *ID2-SP1* | 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) |
| *Parametri* | id, ordine |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *id* | ID1: id presente nel DB e associata ad un ordine [property id\_ok]  ID2: id non presente nel DB [property id\_error] |
| *ordine* | OR1: ordine != null [property ordine\_ok]  OR2: ordine == null || ordine == “”[property ordine\_error] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *ID1-OR1* | Viene aggiornata la descrizione dell’ordine che ha come id quello passato come parametro |
| *ID2-OR1* | 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) |
| *Parametri* | prodotto |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *prodotto* | P1: prodotto != null [property prodotto\_ok]  P2: prodotto == null [property prodotto\_error] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *P1* | 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 |
| *P2* | Viene generato un errore in quanto il parametro prodotto è null |
| *Metodo* | doRetrieveByKey(int id) |
| *Parametri* | id |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *id* | ID1: id presente nel DB e associata ad un ordine [property id\_ok]  ID2: id non presente nel DB [property id\_error] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *ID1* | Viene restituita la lista dei prodotti associati all’ordine che ha come id quello passato come parametro |
| *ID2* | Viene generato un errore in quanto non ci sono corrispondenze all’interno del database dell’id passato come parametro |

**1.3***Recensione*

|  |  |
| --- | --- |
| *RecensioniModelDS* | |
| *Metodo* | doSave(RecensioniBean recensione) |
| *Parametri* | recensione |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *recensione* | R1: recensione != null [property recensione\_ok] R2: recensione == null [property recensione\_error] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *R1* | Viene salvata all’interno del database un’istanza di RecensioneBean, ossia una nuova recensione e associata all’account che l’ha creata |
| *R2* | Non viene creata una nuova istanza di recensione in quanto il parametro è null |
| *Metodo* | doDelete(int numero) |
| *Parametri* | numero |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *numero* | N1: numero presente nel DB e associata ad un ordine [property number\_ok]  N2: numero non presente nel DB [property number\_error] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *N1* | Viene eliminato la recensione che ha come numero quello passato come parametro esplicito |
| *N2* | Viene generato un errore in quanto non ci sono corrispondenze all’interno del database del numero passato come parametro |
| *Metodo* | doRetrieveByKey(String nomeprodotto) |
| *Parametri* | nomeprodotto |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *nomeprodotto* | NP1: nomeprodotto != null e associato ad un prodotto nel database [property number\_ok]  NP2: nomeprodotto != null ma non associata a nessun prodotto [property number\_notexists] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *NP1* | Viene restituita la recensione associata al prodotto che ha come nome quello passato come parametro |
| *NP2* | Viene restituito null in quanto non ci sono prodotti che hanno come nome quello passato come parametro |
| *Metodo* | doRetrieveAll(String nome) |
| *Parametri* | nome |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *nome* | N1: nome != null e associato a prodotto presente nel DB [property nome\_ok]  N2: nome != null e associato a nessun prodotto nel DB [property nome\_notexists]  N3: nome == null || nome == “” [property nome\_error] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *N1* | Viene restituita una lista di RecensioniBean associata al prodotto che ha come nome quello passato come parametro |
| *N2* | 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 |
| *N3* | Viene generato un errore in quanto il parametro nome è null |

**1.4***Prodotto*

|  |  |
| --- | --- |
| *ProdottoModelDS* | |
| *Metodo* | doSave(ProdottoBean prodotto, String email) |
| *Parametri* | prodotto, email |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *prodotto* | P1: prodotto != null [property prodotto\_ok]  P2: prodotto == null [property prodotto\_error] |
| *email* | EM1: email != null [property email\_ok] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *P1-E1* | 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 |
| *P2-E1* | Non viene creata una nuova istanza di prodotto in quanto il parametro prodotto è null |
| *Metodo* | doRetrieveByKey(int code) |
| *Parametri* | code |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *code* | C1: codice presente nel DB e associata ad un prodotto [property code\_ok]  C2: codice non presente nel DB [property code\_error] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *C1* | Viene restituita un’istanza di ProdottoBean del prodotto che ha come codice quello passato come parametro |
| *C2* | 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) |
| *Parametri* | code, email |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *code* | C1: code != null e associato a prodotto nel database [property code\_ok]  C2: code != e associato a nessun prodotto nel database [property code\_error] |
| *email* | EM1: email != null [property email\_ok] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *C1-EM1* | Viene eliminato il prodotto dal database che ha come codice quello passato come parametro |
| *C2-EM1* | Viene generato un errore in quanto non è presente nessun prodotto ha come codice quello passato come parametro |
| *Metodo* | doRetrieveAll(String ordine) |
| *Parametri* | ordine |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *ordine* | OR1: ordine != null [property ordine\_ok]  OR2: ordine != null ma non pertinente [property ordine\_error] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *OR1* | Viene restituita una lista di ProdottoBean contenente i prodotti che sono pertinenti (ORDER BY order) al parametro order |
| *OR2* | Viene restituita una lista vuota in quanto il parametro ordine non è pertinente |
| *Metodo* | doUpdatePrice(int code, int prezzo) |
| *Parametri* | code, prezzo |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *code* | C1: code != null e associato a prodotto nel database [property code\_ok]  C2: code != e associato a nessun prodotto nel database [property code\_error] |
| *prezzo* | P1: prezzo != null [property price\_ok] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *C1-P1* | Viene aggiornato il prezzo con il parametro prezzo passato come parametro del prodotto che ha come codice quello passato come parametro |
| *C2-P1* | Viene generato un errore in quanto il prodotto non è presente nel DB |
| *Metodo* | doRetrieveSearch(String ricerca) |
| *Parametri* | ricerca |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *ricerca* | R1: ricerca != null e pertinente a prodotti nel DB [property ricerca\_ok]  R2: ricerca != null e non pertinente a prodotti nel DB [property ricerca\_not exists] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *R1* | Viene restituita una lista di ProdottoBean in base alla ricerca passata come parametro, ossia una lista dei prodotti pertinenti alla ricerca |
| *R2* | Viene generato un errore in quanto non ci sono prodotti pertinenti alla ricerca |
| *Metodo* | doRetrieveAllProducts() |
| *Parametri* | // |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *//* | // |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *//* | Viene restituita la lista di tutti i prodotti all’interno del database |
| *Metodo* | doUpdateQuantity(int code, int quantità) |
| *Parametri* | code, quantità |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *code* | C1: code != null e associato a prodotto nel database [property code\_ok]  C2: code != null e associato a nessun prodotto nel database [property code\_error] |
| *quantità* | Q1: quantità != null [property quantity\_ok] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *C1-Q1* | Viene aggiornato il campo quantità del prodotto associato al codice passato come parametro con il parametro quantità |
| *C2-Q1* | Viene generato un errore in quanto il prodotto non è presente nel DB |
| *Metodo* | doIncreaseQuantity(int code, int quantità) |
| *Parametri* | code, quantità |
| *Oggetti d’ambiente* | Database |
| *Categorie* | ***Scelte*** |
| *code* | C1: code != null e associato a prodotto nel database [property code\_ok]  C2: code != e associato a nessun prodotto nel database [property code\_error] |
| *quantità* | Q1: quantità != null [property quantity\_ok] |
| *Test frame* | |
| *Percorso scelte* | ***Oracolo*** |
| *C1-Q1* | Viene aggiornato il campo quantità del prodotto associato al codice passato come parametro con il parametro quantità |
| *C2-Q1* | Viene generato un errore in quanto il prodotto non è presente nel DB |

**1.5***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.