|  |
| --- |
| Università degli Studi di Salerno  Corso di Ingegneria del Software |

Kawaii Comix

Test Plan di Unità

Versione 1.1

**Immagine che contiene fiore, rosa, Elementi grafici, testo

Descrizione generata automaticamente**

Data: 27/01/2024

|  |  |
| --- | --- |
| Nome | Matricola |
| Simone D’Assisi | 0512113584 |
| Davide Del Franco Natale | 0512113233 |
| Giovanni Sicilia | 0512107458 |

**Coordinatore del progetto**

**Partecipanti**

|  |  |
| --- | --- |
| Nome | Matricola |
| Simone D’Assisi | 0512113584 |
| Davide Del Franco Natale | 0512113233 |
| Giovanni Sicilia | 0512107458 |

|  |  |
| --- | --- |
| **Scritto da:** | Giovanni Sicilia |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autore** |
| 27/01/2024 | 1.0 | Prima stesura documento | Giovanni Sicilia |
| 30/01/2024 | 1.1 | Aggiunta Test case d’unità | Simone D’Assisi |
|  |  |  |  |
|  |  |  |  |

Sommario

[1. Introduzione 4](#_Toc157533881)

[2. Relazioni con altri documenti 4](#_Toc157533882)

[3. Componenti da testare 4](#_Toc157533883)

[4. Test Case 4](#_Toc157533884)

[4.1. Package utenteManagement 4](#_Toc157533885)

[4.1.1. TCU1 UserIDS 4](#_Toc157533886)

[4.2. Package catalogoManagement 8](#_Toc157533887)

[4.2.1. TCU2 GenereIDS 8](#_Toc157533888)

[4.2.2. TCU3 CategoriaIDS 8](#_Toc157533889)

[4.2.3. TCU4 ProdottoIDS 9](#_Toc157533890)

[4.2.4. TCU5 GestoreCatalogoIDS 13](#_Toc157533891)

[4.3. Package acquistoManagement 15](#_Toc157533892)

[4.3.1. CarrelloIDS 15](#_Toc157533893)

[4.3.2. OrdineSingoloIDS 17](#_Toc157533894)

[4.3.3. OrdineIDS 19](#_Toc157533895)

[4.3.4. GestoreOrdiniIDS 19](#_Toc157533896)

# Introduzione

L’obiettivo di questo documento è quello di pianificare e progettare i test case di riferimento per l’implementazione dei test d’unità.

# Relazioni con altri documenti

* T.P.: Le informazioni riguardanti l’approccio che verrà utilizzato per il testing sono presenti nel Test Plan
* O.D.D.: faremo riferimento alle interfacce definite nell’O.D.D.

# Componenti da testare

Le componenti che andremo a testare sono:

* Package utenteManagement:
  + UserIDS
* Package catalogoManagement:
  + GenereIDS
  + CategoriaIDS
  + ProdottoIDS
  + GestoreCatalogoIDS
* Package acquistoManagement:
  + CarrelloIDS
  + OrdineSingoloIDS
  + OrdineIDS
  + GestoreOrdiniIDS

# Test Case

## Package utenteManagement

### TCU1 UserIDS

#### TCU1\_1 doSaveUser(user)

|  |  |
| --- | --- |
| **Parametri** | user |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| user | US1: user non vuoto e non null |
| Email dell’account | EP1: email presente nel DB  EP2: email non presente nel DB |
| Password dell’account | PW1: Lunghezza password >=8 |
| Nome dell’account | NA1: nome non vuoto e non null |
| Cognome dell’account | CA1: cognome non vuoto e non null |
| Indirizzo dell’account | IN1: Indirizzo non vuoto e non null |
| Città | CI1: città non vuota e non null |
| Codice postale | CP1: Codice postale non vuoto e non null |
| Provincia | PR1: Provincia non vuota e non null |
| Nazione | NAZ1: Nazione non vuota e non null |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| US1 – FE1 – EP2 – PW1 - NA1 – CA1 – IN1 – CI1 -CP1 - PR1 - NAZ1 | L’account viene salvato nel Database |
| US1 – FE1 – EP1 – PW1 - NA1 – CA1 – IN1 – CI1 -CP1 - PR1 - NAZ1 | L’account non viene salvato nel Database |

#### TCU1\_2 doDeleteUser(id)

|  |  |
| --- | --- |
| **Parametri** | id |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| Id | ID1: id presente nel Database  ID2: id non presente nel Database |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| ID1 | L’account con l’id specificato viene eliminato dal Database. Viene restituito *true* |
| ID2 | Restituisce *false* |

#### TCU1\_3 doUpdateUser(user)

|  |  |
| --- | --- |
| **Parametri** | user |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| user | US1: user non vuoto e non null |
| Email dell’account | EP1: email presente nel DB |
| Password dell’account | PW1: Lunghezza password >=8 |
| Nome dell’account | NA1: nome non vuoto e non null |
| Cognome dell’account | CA1: cognome non vuoto e non null |
| Indirizzo dell’account | IN1: Indirizzo non vuoto e non null |
| Città | CI1: città non vuota e non null |
| Codice postale | CP1: Codice postale non vuoto e non null |
| Provincia | PR1: Provincia non vuota e non null |
| Nazione | NAZ1: Nazione non vuota e non null |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| US1 – EP1 – PW1 - NA1 – CA1 – IN1 – CI1 -CP1 - PR1 - NAZ1 | L’account viene aggiornato nel Database. Viene restituito *true* |

#### TCU1\_4 doUpdateUserPassword(user, password)

|  |  |
| --- | --- |
| **Parametri** | User, password |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| user | US1: user non vuoto e non null |
| Password dell’account | PW1: Lunghezza password >=8 |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| US1 – PW1 | Viene aggiornata la password dell’account Database |

#### TCU1\_5 doRetrieveAllUsers()

|  |  |
| --- | --- |
| **Parametri** |  |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
|  | Viene restituita l’intera lista di user presente nel Database |

#### TCU1\_6 doRetrieveById(id)

|  |  |
| --- | --- |
| **Parametri** | id |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| Id | ID1: id presente nel Database  ID2: id non presente nel Database |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| ID1 | Viene restituito lo user il cui userId è pari a id |
| ID2 | Viene restituito uno user null |

#### TCU1\_7 doRetrieveUser(email, password)

|  |  |
| --- | --- |
| **Parametri** | Email, password |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| Email dell’account | EP1: email presente nel DB  EP2: email non presente nel DB |
| Password dell’account | PWA1: password associata all’email data  PWA2: password non associata all’email data |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| EP1 – PWA1 | Viene effettuato il login dello user con email e password date |
| EP2 | Viene restituito uno user *null* |
| EP1 – PWA2 | Non viene effettuato il login dello user con email data. Viene restituito uno user *null* |

#### TCU1\_8 emailExists(email)

|  |  |
| --- | --- |
| **Parametri** | email |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| Email dell’account | EP1: email presente nel DB  EP2: email non presente nel DB |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| EP1 | L’email è presente nel DB, viene restituito *true* |
| EP2 | L’email non è presente nel DB, viene restituito *false* |

## Package catalogoManagement

### TCU2 GenereIDS

#### TCU2\_1 doRetrieveAll()

|  |  |
| --- | --- |
| **Parametri** |  |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
|  | Viene restituita l’intera lista di generi presente nel Database |

#### TCU2\_2 checkGenereName(nome\_genere)

|  |  |
| --- | --- |
| **Parametri** | Nome\_genere |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| Nome\_genere | NG1: nome genere presente nel DB  NG2: nome genere non presente nel DB |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| NG1 | Viene restituito *true* |
| NG2 | Viene restituito *false* |

### TCU3 CategoriaIDS

#### TCU3\_1 doRetrieveAll()

|  |  |
| --- | --- |
| **Parametri** |  |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
|  | Viene restituita l’intera lista di categorie presente nel Database |

#### TCU3\_2 checkCategoriaName(nome\_\_categoria)

|  |  |
| --- | --- |
| **Parametri** | Nome\_categoria |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| Nome\_categoria | NC1: nome genere presente nel DB  NC2: nome genere non presente nel DB |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| NC1 | Viene restituito *true* |
| NC2 | Viene restituito *false* |

### TCU4 ProdottoIDS

#### TCU4\_1 doSaveProdotto(prodotto)

|  |  |
| --- | --- |
| **Parametri** | prodotto |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| Prodotto | PE1: prodotto non vuoto e non null |
| Isbn | IS1: isbn presente nel DB  IS2: isbn non presente nel DB |
| Nome | NO1: nome non vuoto e non null |
| Autore | AU1: autore non vuoto e non null |
| Descrizione | DE1: descrizione non vuota e non null |
| Immagine | IM1: immagine non vuota e non null |
| Prezzo | PR1: prezzo > 0 |
| Quantità | QU1: quantità > 0 |
| Genere | GE1: genere non vuoto e non null |
| Categoria | CA1: categoria non vuota e non null |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| PE1 – IS1 – NO1 – AU1 – DE1 – IM1 – PR1 – QU1 – GE1 – CA1 | Il prodotto viene salvato nel Database |
| PE1 – IS2 – NO1 – AU1 – DE1 – IM1 – PR1 – QU1 – GE1 – CA1 | Il prodotto non viene salvato nel Database |

#### TCU4\_2 doDeleteProdotto(isbn)

|  |  |
| --- | --- |
| **Parametri** | isbn |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| Isbn | IS1: isbn presente nel DB  IS2: isbn non presente nel DB |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| IS1 | Il prodotto viene eliminato dal Database. Viene restituito *true* |
| IS2 | Viene restituito *false* |

#### TCU4\_3 doUpdateProdotto(prodotto)

|  |  |
| --- | --- |
| **Parametri** | prodotto |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| Prodotto | PE1: prodotto non vuoto e non null |
| Isbn | IS1: isbn presente nel DB  IS2: isbn non presente nel DB |
| Nome | NO1: nome non vuoto e non null |
| Autore | AU1: autore non vuoto e non null |
| Descrizione | DE1: descrizione non vuota e non null |
| Immagine | IM1: immagine non vuota e non null |
| Prezzo | PR1: prezzo > 0 |
| Quantità | QU1: quantità > 0 |
| Genere | GE1: genere non vuoto e non null |
| Categoria | CA1: categoria non vuota e non null |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| PE1 – IS2 – NO1 – AU1 – DE1 – IM1 – PR1 – QU1 – GE1 – CA1 | Il prodotto viene aggiornato nel Database |
| PE1 – IS1 – NO1 – AU1 – DE1 – IM1 – PR1 – QU1 – GE1 – CA1 | Il prodotto non viene aggiornato nel Database |

#### TCU4\_4 doRetrieveAllProdotti()

|  |  |
| --- | --- |
| **Parametri** |  |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
|  | Viene restituita l’intera lista di prodotti presente nel Database |

#### TCU4\_5 doRetrieveByIsbn(isbn)

|  |  |
| --- | --- |
| **Parametri** | isbn |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| Isbn | IS1: isbn presente nel DB  IS2: isbn non presente nel DB |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| IS1 | Viene restituito il prodotto il cui l’isbn è pari a *isbn* |
| IS2 | Viene restituito un prodotto null |

#### TCU4\_6 doRetrieveByNome(nome\_prodotto)

|  |  |
| --- | --- |
| **Parametri** | Nome\_prodotto |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| Nome\_prodotto | NP1: nome\_prodotto presente nel DB  NP2: nome\_prodotto non presente nel DB |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| NP1 | Viene restituito il prodotto il cui nome è pari a nome\_prodotto |
| NP2 | Viene restituito un prodotto null |

#### TCU4\_7 lastSaved()

|  |  |
| --- | --- |
| **Parametri** |  |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
|  | Viene restituita la lista degli ultimi cinque prodotti aggiunti al Database |

#### TCU4\_8 doRetrieveAllProductsName()

|  |  |
| --- | --- |
| **Parametri** |  |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
|  | Viene restituita l’intera lista dei nomi dei prodotti presenti nel Database |

#### TCU4\_9 updateCopieVendute(prodotto)

|  |  |
| --- | --- |
| **Parametri** | prodotto |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| Prodotto | PE1: prodotto non vuoto e non null |
| Copie\_vendute | CV1: copie\_vendute non null |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| PE1– CV1 | Viene aggiornato il campo *copie\_vendute* per il prodotto nel Database |

#### TCU4\_10 bestSellers()

|  |  |
| --- | --- |
| **Parametri** |  |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
|  | Viene restituita la lista dei cinque prodotti il cui campo *copie\_vendute* è massimo |

### TCU5 GestoreCatalogoIDS

#### TCU5\_1 doSaveGestore(gestoreCatalogo)

|  |  |
| --- | --- |
| **Parametri** | gestoreCatalogo |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| gestoreCatalogo | GC1: gestoreCatalogo non vuoto e non null |
| Email | EP1: email presente nel DB  EP2: email non presente nel DB |
| Password | PW1: Lunghezza password >=8 |
| Nome | NA1: nome non vuoto e non null |
| Cognome | CA1: cognome non vuoto e non null |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| GC1 – EP2 – PW1 - NA1 – CA1 | Il Gestore del Catalogo viene salvato nel Database |
| GC1 – EP1 – PW1 - NA1 – CA1 | Il Gestore del Catalogo non viene salvato nel Database |

#### TCU5\_2 doUpdateGestore(gestoreCatalogo)

|  |  |
| --- | --- |
| **Parametri** | gestoreCatalogo |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| gestoreCatalogo | GC1: gestoreCatalogo non vuoto e non null |
| Email | EP1: email presente nel DB  EP2: email non presente nel DB |
| Password | PW1: Lunghezza password >=8 |
| Nome | NA1: nome non vuoto e non null |
| Cognome | CA1: cognome non vuoto e non null |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| GC1 – EP1 – PW1 - NA1 – CA1 | Il Gestore del Catalogo viene aggiornato nel Database |
| GC1 – EP2 – PW1 - NA1 – CA1 | Il Gestore del Catalogo non viene aggiornato nel Database |

#### TCU5\_3 doDeleteGestore(email)

|  |  |
| --- | --- |
| **Parametri** | email |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| Email | EP1: email presente nel DB  EP2: email non presente nel DB |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| EP1 | Il Gestore del Catalogo viene eliminato dal Database. Viene restituito *true* |
| EP2 | Viene restituito *false* |

#### TCU5\_4 doRetrieveByAuthentication(email, password)

|  |  |
| --- | --- |
| **Parametri** | Email, password |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| Email | EP1: email presente nel DB  EP2: email non presente nel DB |
| Password | PWA1: password associata all’email data  PWA2: password non associata all’email data |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| EP1 – PWA1 | Viene effettuato il login del Gestore del Catalogo con email e password date |
| EP2 | Viene restituito un Gestore del Catalogo *null* |
| EP1 – PWA2 | Non viene effettuato il login del Gestore del Catalogo con email data. Viene restituito un Gestore del Catalogo *null* |

## Package acquistoManagement

### CarrelloIDS

#### TCU6\_1 doRetrieveCarrello(userId)

|  |  |
| --- | --- |
| **Parametri** | userId |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| userId | UID1: carrello presente per lo userId  UID2: carrello non presente per lo userId |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| UID1 | Viene restituito il carrello il cui lo userId è pari a *userId* |
| UID2 | Viene restituito un carrello null |

#### TCU6\_2 doRetrieveProdottiCarrello(carrello)

|  |  |
| --- | --- |
| **Parametri** | carrello |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| carrello | CA1: carrello non vuoto e non null |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| CA1 | Viene restituita la lista prodotti associata al carrello |

#### TCU6\_3 doDeleteProdottiCarrello(carrello)

|  |  |
| --- | --- |
| **Parametri** | carrello |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| carrello | CA1: carrello non vuoto e non null |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| CA1 | Svuota la lista prodotti associata al carrello |

#### TCU6\_4 doCreareCarrello(userId)

|  |  |
| --- | --- |
| **Parametri** | userId |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| userId | UID1: userId presente nel Database  UID2: userId non presente nel Database |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| UID1 | Viene creato nel Database un carrello per l’utente con *userId* |
| UID2 | Non viene creato nel Database un carrello per l’utente con *userId* |

#### TCU6\_5 doDeleteCarrello(carrelloId)

|  |  |
| --- | --- |
| **Parametri** | carrelloId |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| carrelloId | CID1: carrelloId presente nel DB  CID2: carrelloId non presente nel DB |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| CID1 | Il carrello viene eliminato dal Database. Viene restituito *true* |
| CID2 | Viene restituito *false* |

#### TCU6\_6 doSvuotaCarrello(carrello)

### OrdineSingoloIDS

#### TCU7\_1 doSaveOrdineSingolo(ordineSingolo)

|  |  |
| --- | --- |
| **Parametri** | ordineSingolo |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| ordineSingolo | OS1: ordineSingolo non vuoto e non null |
| Id | ID1: id presente nel DB  ID2: id non presente nel DB |
| Quantità | QU1: nome non vuoto e non null |
| totParziale | TP1: autore non vuoto e non null |
| ordineId | OID1: descrizione non vuota e non null |
| Prodotto | PR1: immagine non vuota e non null |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| OS1 – ID2 – QU1 – TP1 – OID1 – PR1 | L’ordine Singolo viene salvato nel Database |
| OS1 – ID1 – QU1 – TP1 – OID1 – PR1 | L’ordine Singolo non viene salvato nel Database |

#### TCU7\_2 doDeleteOrdineSingolo(id)

|  |  |
| --- | --- |
| **Parametri** | id |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| Id | ID1: id presente nel DB  ID2: id non presente nel DB |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| ID1 | I’ordine Singolo viene eliminato dal Database. Viene restituito *true* |
| ID2 | Viene restituito *false* |

#### TCU7\_3 doUpdateOrdineSingolo(ordineSingolo)

|  |  |
| --- | --- |
| **Parametri** | ordineSingolo |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| ordineSingolo | OS1: ordineSingolo non vuoto e non null |
| Id | ID1: id presente nel DB  ID2: id non presente nel DB |
| Quantità | QU1: quantità non vuoto e non null |
| totParziale | TP1: totParziale non vuoto e non null |
| ordineId | OID1: ordineId non vuota e non null |
| Prodotto | PR1: prodotto non vuota e non null |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| OS1 – ID1 – QU1 – TP1 – OID1 – PR1 | L’ordine Singolo viene aggiornato nel Database |
| OS1 – ID2 – QU1 – TP1 – OID1 – PR1 | L’ordine Singolo non viene aggiornato nel Database |

#### TCU7\_4 doRetrieveAllOrdineSingolo()

|  |  |
| --- | --- |
| **Parametri** |  |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
|  | Viene restituita l’intera lista di ordini Singoli presente nel Database |

#### TCU7\_5 doRetrieveAllByOrdineId(ordineId)

|  |  |
| --- | --- |
| **Parametri** | ordineId |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| ordineId | OID1: ordineId non vuoto e non null |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
|  | Viene restituita l’intera lista di ordini Singoli con il campo ordine\_id pari a *ordineId* |

#### TCU7\_6 doRetrieveById(id)

|  |  |
| --- | --- |
| **Parametri** | id |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| Id | ID1: id non vuoto e non null |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
|  | Viene restituita l’ordine Singolo con il campo id pari a *id* |

### OrdineIDS

#### TCU8\_1 doSaveOrdine (ordine)

#### TCU8\_2 doDeleteOrdine(id)

#### TCU8\_3 doUpdateOrdine (ordine)

#### TCU8\_4 doUpdateStatoById (id, stato)

#### TCU8\_5 doRetrieveAllOrdini()

#### TCU8\_6 doRetrieveById(id)

#### TCU8\_7 doRetrieveByUserId(userId)

#### TCU8\_8 doRetrieveAllOrdiniSingoli(ordine)

#### TCU8\_9 doSaveOrdineSingoloAssociato(ordineSingolo)

### GestoreOrdiniIDS

#### TCU9\_1 doSaveGestore(gestoreCatalogo)

|  |  |
| --- | --- |
| **Parametri** | gestoreOridni |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| gestoreCatalogo | GC1: gestoreOridni non vuoto e non null |
| Email | EP1: email presente nel DB  EP2: email non presente nel DB |
| Password | PW1: Lunghezza password >=8 |
| Nome | NA1: nome non vuoto e non null |
| Cognome | CA1: cognome non vuoto e non null |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| GC1 – EP2 – PW1 - NA1 – CA1 | Il Gestore degli Ordini viene salvato nel Database |
| GC1 – EP1 – PW1 - NA1 – CA1 | Il Gestore degli Ordini non viene salvato nel Database |

#### TCU5\_2 doUpdateGestore(gestoreCatalogo)

|  |  |
| --- | --- |
| **Parametri** | gestoreOridni |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| gestoreCatalogo | GC1: gestoreOridni non vuoto e non null |
| Email | EP1: email presente nel DB  EP2: email non presente nel DB |
| Password | PW1: Lunghezza password >=8 |
| Nome | NA1: nome non vuoto e non null |
| Cognome | CA1: cognome non vuoto e non null |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| GC1 – EP1 – PW1 - NA1 – CA1 | Il Gestore degli Ordini viene aggiornato nel Database |
| GC1 – EP2 – PW1 - NA1 – CA1 | Il Gestore degli Ordini non viene aggiornato nel Database |

#### TCU5\_3 doDeleteGestore(email)

|  |  |
| --- | --- |
| **Parametri** | email |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| Email | EP1: email presente nel DB  EP2: email non presente nel DB |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| EP1 | Il Gestore degli Ordini viene eliminato dal Database. Viene restituito *true* |
| EP2 | Viene restituito *false* |

#### TCU5\_4 doRetrieveByAuthentication(email, password)

|  |  |
| --- | --- |
| **Parametri** | Email, password |
| **Oggetti dell’ambiente** | Database |
| **Categorie** | **Scelte** |
| Email | EP1: email presente nel DB  EP2: email non presente nel DB |
| Password | PWA1: password associata all’email data  PWA2: password non associata all’email data |

**Test Frame**

|  |  |
| --- | --- |
| **Combinazione** | **Oracolo** |
| EP1 – PWA1 | Viene effettuato il login del Gestore degli Ordini con email e password date |
| EP2 | Viene restituito un Gestore degli Ordini *null* |
| EP1 – PWA2 | Non viene effettuato il login del degli Ordini con email data. Viene restituito un degli Ordini *null* |