



iSnapGaming

TP

| | |
|---|-----------|
| Test Plan | 4 |
| Introduction | 5 |
| Relationship to other documents | 5 |
| System Overview | 5 |
| Features to be tested/not to be tested | 5 |
| Pass/Fail criteria | 6 |
| Approach | 6 |
| Tools used for testing | 6 |
| Test Cases | 6 |
| Funcionality Testing | 6 |
| UserManagement: | 6 |
| UserManagement | 11 |
| User | 11 |
| makeUser | 11 |
| Address | 12 |
| makeAddress | 12 |
| OrderManager | 12 |
| getAllCustomerOrders | 12 |
| checkProduct | 13 |
| packProduct | 13 |
| replaceProduct | 14 |
| contactCourier | 14 |
| restoreOrder | 14 |
| confirmDelivery | 15 |
| ProductManager | 15 |
| addProduct | 15 |
| removeProduct | 16 |
| updateProduct | 16 |
| makeProductAvailable | 17 |
| getProductByProdCode | 17 |
| ProductManagement | 18 |
| Product | 18 |
| makeProduct | 18 |
| OrderManagement | 19 |
| Cart | 19 |
| addToCart | 19 |
| removeFromCart | 20 |
| decreaseQuantityCart | 20 |
| CustomerOrder | 21 |
| makeCustomerOrder | 21 |
| addProduct | 22 |
| OrderCreation | 22 |
| makeOrder | 22 |

| | |
|------------------------------------|-----------|
| PaymentManagement | 23 |
| PaymentAuthorizationServiceAdapter | 23 |
| checkPaymentData | 23 |
| StorageManagement | 24 |
| AddressDAO | 24 |
| doSave | 24 |
| findByKey | 24 |
| findByCustomerId | 25 |
| ProductDAO | 25 |
| doSave | 25 |
| doUpdate | 26 |
| findByKey | 26 |
| findByProdCode | 27 |
| findByCategory | 27 |
| doRetrieveAll | 28 |
| UserDAO | 28 |
| doSave | 28 |
| getUserByUsernameAndPassword | 28 |
| getUserRoles | 29 |
| findByKey | 30 |
| findByUsername | 30 |
| CustomerDAO | 31 |
| doSave | 31 |
| findByKey | 31 |
| findAddressByCustomerId | 31 |
| CustomerOrderDAO | 32 |
| doSave | 32 |
| updateStatus | 32 |
| findByKey | 33 |
| doRetrieveAll | 33 |
| Test schedule | 34 |

Test Plan

Introduction

Il presente documento ha l'obiettivo di specificare la pianificazione dell'attività di testing per la piattaforma iSnapGaming.

Verranno riportati gli elementi e le funzionalità da testare, le strategie di testing e gli strumenti utilizzati.

Relationship to other documents

- **RAD**: i test cases sono definiti in relazione ai requisiti funzionali e non funzionali descritti nel RAD.
- **SDD**: i test cases rispettano la suddivisione in sottosistemi riportata nell'SDD.
- **ODD**: elementi e funzionalità da testare fanno riferimento alle specifiche dell'ODD.

System Overview

L'architettura del sistema è di tipo three-tier, dunque presenta i seguenti livelli:

- Presentation Layer
- Business Layer
- Data Layer

In oltre, come riportato nell'SDD, il sistema è stato diviso nei seguenti sottosistemi:

- UserManagement Subsystem
- ProductManagement Subsystem
- OrderManagement Subsystem
- PaymentManagement Subsystem
- Storage Management Subsystem

Features to be tested/not to be tested

Verranno testate le seguenti funzionalità per ogni sottosistema:

UserManagement:

- Registrazione
- Autenticazione
- Cambio ruolo

ProductManagement:

- Aggiunta di un prodotto
- Modifica di un prodotto
- Rimozione di un prodotto

OrderManagement:

- Aggiunta di un ordine
- Aggiornamento stato di un ordine
- Aggiunta di un prodotto al carrello
- Rimozione di un prodotto dal carrello

PaymentManagement:

- Acquisto di un prodotto

Pass/Fail criteria

In caso non si riscontrino failures in un test, esso verrà considerato **passed**. Altrimenti, in caso si riscontrino una o più failures, il test verrà considerato **failed**.

Approach

L'approccio considera tre fasi che si susseguiranno nel seguente ordine:

- Test di unità per le singole parti del sistema;
- Test di integrazione;
- Test di sistema per testare tutte le funzionalità implementate nel sistema nel loro complesso.

Tools used for testing

- JUnit: per realizzare il testing di unità e quello di integrazione.
- Mockito: utile nel test di unità e in quello di integrazione, per "simulare" le dipendenze di un classe verso un'altra.
- Selenium: per automatizzare il testing funzionale.

Test Cases

Functionality Testing

UserManagement:

Autenticazione

| | |
|----------------------------|--------------------------|
| PARAMETERS | username, password, role |
| OBJECTS OF THE ENVIRONMENT | Database |
| CATEGORIES | CHOICES |

| | |
|--|---|
| Username presente nel database | UP1: username presente nel database [property username_presente] UP2: username non presente nel database |
| Password presente nel database | PP1: password associata a username nel database [property password_associata] PP2: password non associata a username nel database [if username_presente] |
| Ruolo associato a username presente nel database | RP1: ruolo associato a username nel database RP2: ruolo non associato a username nel database [if password_presente & password_associata] |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|--|
| UP1-PP1-RP1 | Utente autenticato con il ruolo scelto e reindirizzato alla corrispondente homepage. |
| UP2-PP1-RP1 | Visualizzazione errore "Wrong username or password" |
| UP1-PP2-RP1 | Visualizzazione errore "Wrong username or password" |
| UP1-PP1-RP2 | Visualizzazione errore "Invalid role" |

Registrazione

| | |
|----------------------------|--|
| PARAMETERS | firstName, lastName, dateOfBirth, email, password, conferma password, street, city, postalCode |
| OBJECTS OF THE ENVIRONMENT | Database |
| CATEGORIES | CHOICES |
| Formato firstName | F1: firstName rispetta il formato del nome F2: firstName non rispetta il formato del nome |
| Formato lastName | L1: lastName rispetta il formato del cognome L2: lastName non rispetta il formato del cognome |
| Valore dateOfBirth | D1: D2: |
| Formato email | E1: lastName rispetta il formato del cognome E2: lastName non rispetta il formato del cognome |
| Formato password | P1: password rispetta il formato password P2: password non rispetta il formato password |

| | |
|------------------------|--|
| Valore confirmPassword | CP1: Conferma password corrisponde a password CP2: Conferma password non corrisponde a password |
| valore Street | S1: S2: |
| Formato city | C1: city rispetta il formato city C2: city non rispetta il formato city |
| Formato postalCode | PC1: postalCode rispetta il formato postalCode PC2: postalCode non rispetta il formato postalCode |

TEST CASES

| COMBINATIONS | ORACLE |
|------------------------------|---|
| F1-L1-D1-E1-P1-CP1-S1-C1-PC1 | Utente registrato correttamente. |
| F1-L1-D1-E1-P1-CP1-S1-C1-PC2 | Visualizzazione errore "The postal code must have only 5 numbers" |
| F1-L1-D1-E1-P1-CP1-S1-C2-PC1 | Visualizzazione errore "The city must contain only letters" |
| F1-L1-D1-E1-P1-CP1-S2-C1-PC1 | Visualizzazione errore "Fill in this field" |
| F1-L1-D1-E1-P1-CP2-S1-C1-PC1 | Visualizzazione errore "Passwords do not match" |
| F1-L1-D1-E1-P2-CP1-S1-C1-PC1 | Visualizzazione errore "Password must contain at least one number, one uppercase and lowercase letter, and at least 8 or more characters" |
| F1-L1-D1-E2-P1-CP1-S1-C1-PC1 | Visualizzazione errore "Email entered not in correct format" |
| F1-L1-D2-E1-P1-CP1-S1-C1-PC1 | Visualizzazione errore "Fill in this field" |
| F1-L2-D1-E1-P1-CP1-S1-C1-PC1 | Visualizzazione errore "Last name must contain only letters" |
| F2-L1-D1-E1-P1-CP1-S1-C1-PC1 | Visualizzazione errore "First name must contain only letters" |

Unit Testing

UserManagement

User

makeUser

| METHOD NAME | makeUser |
|--------------------|---|
| PARAMETERS | username: String, password: String, firstName: String, lastName: String, dateOfBirth: LocalDate |
| CATEGORIES | CHOICES |
| Valore username | U1:username!= null U2:username=null |
| valore password | P1:valore password corretto P2:valore password corretto |
| valore firstName | F1:firstName!=null F2:firstName=null |
| valore lastName | L1:lastName!=null L2:lastName=null |
| valore dateOfBirth | B1:dateOfBirth!=null B2:dateOfBirth=null |

| COMBINATIONS | ORACLE |
|----------------|---|
| U1-P1-F1-L1-B1 | Corretto:viene creato un user con i campi specificati |
| U1-P1-F1-L1-B2 | Error:"Date of Birth cannot be null" |
| U1-P1-F1-L2-B1 | Error:"Last Name cannot be null" |
| U1-P1-F2-L1-B1 | Error:"First Name cannot be null" |
| U1-P2-F1-L1-B1 | Error:"Password cannot be null" |
| U2-P1-F1-L1-B1 | Error:"Username cannot be null" |

Address

makeAddress

| METHOD NAME | makeAddress |
|-------------------|--|
| PARAMETERS | customerId: Integer, street: String, city: String, postalCode: Integer |
| CATEGORIES | CHOICES |
| Valore customerId | C1:valore customerId>=0 C2:valore customerId>0 |
| valore street | S1:street!=null S2:street=null |
| valore city | CT1:city!=null CT2:city=null |
| valore postalCode | P1:valore PostalCode>0 P2:valore PostalCode<0 |

| COMBINATIONS | ORACLE |
|--------------|--|
| C1-S1-CT1-P1 | Corretto:viene creato un address con i campi specificati |
| C1-S1-CT1-P2 | Error:"Postal code cannot be negative" |
| C1-S1-CT2-P1 | Error:"City cannot be null" |
| C1-S2-CT1-P1 | Error:"Street cannot be null" |
| C2-S1-CT1-P1 | Error:"CustomerId cannot be negative" |

OrderManager

getAllCustomerOrders

| METHOD NAME | getAllCustomerOrders |
|---------------------|------------------------|
| PARAMETERS | dataSource: DataSource |
| ENVIRONMENT OBJECTS | |
| CATEGORIES | CHOICES |

| | |
|-----------------|---|
| Presenza ordini | OP1: zero corrispondenze OP2: una o più corrispondenze |
|-----------------|---|

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|---|
| OP1 | OK: viene restituita una lista vuota |
| OP2 | OK: viene restituita una lista con uno o più ordini |

checkProduct

| | |
|---------------------|--|
| METHOD NAME | checkProduct |
| PARAMETERS | order: CustomerOrder, dataSource: DataSource |
| ENVIRONMENT OBJECTS | |
| CATEGORIES | CHOICES |
| valore order | O1: valore !=null O2: valore =null |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|-------------------------------------|
| O1 | OK |
| O2 | Error:”Cannot have empty arguments” |

packProduct

| | |
|---------------------|--|
| METHOD NAME | packProduct |
| PARAMETERS | order: CustomerOrder, dataSource: DataSource |
| ENVIRONMENT OBJECTS | |
| CATEGORIES | CHOICES |
| valore order | O1: valore !=null O2: valore =null |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|------------------------------|
| O1 | OK |
| O2 | Error:”Order cannot be null” |

replaceProduct

| | |
|----------------------------|--|
| METHOD NAME | replaceProduct |
| PARAMETERS | order: CustomerOrder, dataSource: DataSource |
| ENVIRONMENT OBJECTS | |
| CATEGORIES | CHOICES |
| valore order | O1: valore !=null O2: valore =null |

TEST CASES

| | |
|---------------------|------------------------------|
| COMBINATIONS | ORACLE |
| O1 | OK |
| O2 | Error:"Order cannot be null" |

contactCourier

| | |
|----------------------------|--|
| METHOD NAME | contactCourier |
| PARAMETERS | order: CustomerOrder, dataSource: DataSource |
| ENVIRONMENT OBJECTS | |
| CATEGORIES | CHOICES |
| valore order | O1: valore !=null O2: valore =null |

TEST CASES

| | |
|---------------------|------------------------------|
| COMBINATIONS | ORACLE |
| O1 | OK |
| O2 | Error:"Order cannot be null" |

restoreOrder

| | |
|----------------------------|--|
| METHOD NAME | restoreOrder |
| PARAMETERS | order: CustomerOrder, dataSource: DataSource |
| ENVIRONMENT OBJECTS | |
| CATEGORIES | CHOICES |
| valore order | O1: valore !=null O2: valore =null |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|------------------------------|
| O1 | OK |
| O2 | Error:"Order cannot be null" |

confirmDelivery

| | |
|---------------------|--|
| METHOD NAME | confirmDelivey |
| PARAMETERS | order: CustomerOrder, dataSource: DataSource |
| ENVIRONMENT OBJECTS | |
| CATEGORIES | CHOICES |
| valore order | O1: valore !=null O2: valore =null |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|------------------------------|
| O1 | OK |
| O2 | Error:"Order cannot be null" |

ProductManager

addProduct

| | |
|---------------------|--|
| METHOD NAME | addProduct |
| PARAMETERS | product: Product, dataSource: DataSource |
| ENVIRONMENT OBJECTS | |
| CATEGORIES | CHOICES |
| Valore product | A1: valore NON null [property productNotNull] A2: valore null |

Test Cases

| COMBINATIONS | ORACLE |
|--------------|---------------------------------|
| A1 | OK |
| A2 | Error: "Product cannot be null" |

removeProduct

| | |
|----------------------------|--|
| METHOD NAME | removeProduct |
| PARAMETERS | product: Product, dataSource: DataSource |
| ENVIRONMENT OBJECTS | |
| CATEGORIES | CHOICES |
| Valore product | A1: valore NON null [property productNotNull] A2: valore null |

Test Cases

| | |
|---------------------|----------------------------------|
| COMBINATIONS | ORACLE |
| A1 | OK |
| A2 | Error: "Product cannot be null." |

updateProduct

| | |
|----------------------------|--|
| METHOD NAME | updateProduct |
| PARAMETERS | product: Product, dataSource: DataSource |
| ENVIRONMENT OBJECTS | |
| CATEGORIES | CHOICES |
| Valore product | A1: valore NON null [property productNotNull] A2: valore null |

Test Cases

| | |
|---------------------|----------------------------------|
| COMBINATIONS | ORACLE |
| A1 | OK |
| A2 | Error: "Product cannot be null." |

makeProductAvailable

| | |
|----------------------------|--|
| METHOD NAME | makeProductAvailable |
| PARAMETERS | product: Product, dataSource: DataSource |
| ENVIRONMENT OBJECTS | |
| CATEGORIES | CHOICES |
| Valore product | A1: valore NON null [property productNotNull] A2: valore null |

Test Cases

| | |
|---------------------|----------------------------------|
| COMBINATIONS | ORACLE |
| A1 | OK |
| A2 | Error: "Product cannot be null". |

getProductByProdCode

| | |
|----------------------------|---|
| METHOD NAME | getProductByProdCode |
| PARAMETERS | prodCode: Integer, dataSource: DataSource |
| ENVIRONMENT OBJECTS | |
| CATEGORIES | CHOICES |
| Valore prodCode | A1: valore NON null [property prodCodeNotNull] A2: valore null |

Test Cases

| | |
|---------------------|----------------------------|
| COMBINATIONS | ORACLE |
| A1 | OK |
| A2 | Error: "Code cannot be 0". |

ProductManagement

Product

makeProduct

| METHOD NAME | makeProduct |
|-----------------------|--|
| PARAMETERS | prodCode: Integer, name: String, softwareHouse: String, platform: Product.Platform, price: Integer, quantity: Integer, category: Product.Category, pegi: Product.Pegi, releaseYear: Integer, imagePath: String |
| CATEGORIES | CHOICES |
| Valore prodCode | P1: valore>0 P2: valore=0 |
| valore name | N1:name != null N2:name =null |
| valore software House | SH1:software!=null SH2:software=null |
| valore platform | PL1:platform!=null PL2:platform=null |
| valore price | PR1:valore>0 PR2: valore<0 |
| valore quantity | Q1:valore>0 Q2: valore<0 |
| valore category | C1:category!=null C2:category=null |
| valore pegi | PE1:pegi!=null PE2:pegi=null |
| valore release Year | RY1:valore>0 RY2:valore<0 |
| valore imagePath | IP1:imagePath !=null IP2: imagePath= null |

| COMBINATIONS | ORACLE |
|-------------------------------------|---|
| P1-N1-SH1-PL1-PR1-Q1-C1-PE1-RY1-IP1 | Corretto:viene creato il prodotto con i campi specificati |

| | |
|-------------------------------------|---|
| P1-N1-SH1-PL1-PR1-Q1-C1-PE1-RY1-IP2 | Corretto:viene creato il prodotto con i campi specificati |
| P1-N1-SH1-PL1-PR1-Q1-C1-PE1-RY2-IP1 | Error:"Release Date cannot be negative" |
| P1-N1-SH1-PL1-PR1-Q1-C1-PE2-RY1-IP1 | Error:"Pegi cannot be null". |
| P1-N1-SH1-PL1-PR1-Q1-C2-PE1-RY1-IP1 | Error:"Category cannot be null". |
| P1-N1-SH1-PL1-PR1-Q2-C1-PE1-RY1-IP1 | Error:"Quantity cannot be negative". |
| P1-N1-SH1-PL1-PR2-Q-C1-PE1-RY1-IP1 | Error:"Price cannot be negative". |
| P1-N1-SH1-PL2-PR1-Q-C1-PE1-RY1-IP1 | Error:"Platform cannot be null". |
| P1-N1-SH2-PL1-PR1-Q-C1-PE1-RY1-IP1 | Error:"Software House cannot be null". |
| P1-N2-SH1-PL1-PR1-Q-C1-PE1-RY1-IP1 | Error:"Name cannot be null". |
| P2-N1-SH1-PL1-PR1-Q-C1-PE1-RY1-IP1 | Error:"Code cannot be 0". |

OrderManagement

Cart

addToCart

| | |
|----------------------------|--|
| METHOD NAME | addToCart |
| PARAMETERS | product: Product, quantity: Integer |
| ENVIRONMENT OBJECTS | |
| CATEGORIES | CHOICES |
| Valore product | A1: valore NON null [property productNotNull] A2: valore null |
| Valore quantity | B1: valore > 0 [property validQuantity] B2: valore <= 0 [if productNotNull] |
| Stato carrello | C1: prodotto Non presente nel carrello [if productNotNull & validQuantity] C2: prodotto presente nel carrello [if productNotNull & validQuantity] |

TEST CASES

| | |
|---------------------|---------------|
| COMBINATIONS | ORACLE |
|---------------------|---------------|

| | |
|----------|--|
| A1-B1-C1 | OK |
| A2-B1-C1 | Error: "Product cannot be null" |
| A1-B2-C1 | Error: "Quantity must be greater than 0" |
| A1-B1-C2 | OK |

removeFromCart

| | |
|---------------------|--|
| METHOD NAME | removeFromCart |
| PARAMETERS | product: Product |
| ENVIRONMENT OBJECTS | |
| CATEGORIES | CHOICES |
| Valore product | A1: valore NON null [property productNotNull] A2: valore null |
| Stato carrello | B1: prodotto presente nel carrello [if productNotNull] B2: prodotto NON presente nel carrello [if productNotNull] |

TEST CASES

| | |
|--------------|---|
| COMBINATIONS | ORACLE |
| A1-B1 | OK |
| A2-B1 | Error: "Product cannot be null" |
| A1-B2 | Error: "No such product found in cart." |

decreaseQuantityCart

| | |
|---------------------|--|
| METHOD NAME | decreaseQuantityCart |
| PARAMETERS | product: Product, quantity: Integer |
| ENVIRONMENT OBJECTS | |
| CATEGORIES | CHOICES |
| Valore product | A1: valore NON null [property productNotNull] A2: valore null |

| | |
|-----------------|--|
| Stato carrello | B1: prodotto presente nel carrello [if productNotNull] [property prodPresente] B2: prodotto NON presente nel carrello [if productNotNull] [property productNonPresente] |
| Valore quantity | C1: valore > 0 & valore < quantità prodotto già inserito [property validQuantity] [if prodPresente] C2: valore <= 0 [if productNotNull] C3: valore = quantità prodotto già inserito [if productPresente] C4: valore > quantità prodotto già inserito [if productPresente productNonPresente] |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|--|
| A1-B1-C1 | OK |
| A2-B1-C1 | Error: "Product cannot be null" |
| A1-B2-C4 | Error: "No such product found in cart." |
| A1-B1-C2 | Error: "Quantity must be greater than 0." |
| A1-B1-C4 | Error: "Invalid quantity" |
| A1-B1-C3 | Ok: prodotto rimosso correttamente dal carrello. |

CustomerOrder

makeCustomerOrder

| | |
|-------------------|--|
| METHOD NAME | makeCustomerOrder |
| PARAMETERS | customerId: Integer, address: String, orderDate: LocalDate, products: List<OrderProduct> |
| CATEGORIES | CHOICES |
| valore customerId | C1:valore>0 C2:valore<0 |
| valore address | A1:address!=null A2:address=null |
| valore orderDate | O1:orderDate!=null O2:orderDate=null |
| valore products | P1:products!=null P2:products=null |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|--|
| C1-A1-O1-P1 | Corretto:viene creato un customerOrder con i campi specificati |
| C1-A1-O1-P2 | Error: "Cannot have an order with no products in it." |
| C1-A1-O2-P1 | Error: "OrderDate cannot be null". |
| C1-A2-O1-P1 | Error: "Address cannot be null". |
| C2-A1-O1-P1 | Error: "CustomerId cannot be negative". |

addProduct

| METHOD NAME | addProduct |
|----------------|---------------------------------------|
| PARAMETERS | product: Product |
| CATEGORIES | CHOICES |
| valore product | P1: product!=null P2: product=null |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|---------------------------------|
| P1 | OK |
| P2 | Error: "Product cannot be null" |

OrderCreation

makeOrder

| METHOD NAME | makeOrder |
|---------------------|---|
| PARAMETERS | cart: Cart, customer: User, address: String |
| ENVIRONMENT OBJECTS | |
| CATEGORIES | CHOICES |
| Valore cart | CA1: cart non null [property validCart] CA2: cart null |
| Valore customer | CU1: customer non null [property validCustomer] |

| | |
|----------------|---|
| | CU2: customer null [if validCustomer] |
| Valore address | A1: address non null A2: address null [if validCart & validCustomer] |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|---|
| CA1-CU1-A1 | OK |
| CA2-CU1-A1 | Error: "There was an error in creating the order" |
| CA1-CU2-A1 | Error: "There was an error in creating the order" |
| CA1-CU1-A2 | Error: "There was an error in creating the order" |

PaymentManagement

PaymentAuthorizationServiceAdapter

checkPaymentData

| | |
|---------------------|--|
| METHOD NAME | checkPaymentData |
| PARAMETERS | cardNumber: String, expiration: LocalDate, cvv: String, price: Integer |
| ENVIRONMENT OBJECTS | |
| CATEGORIES | CHOICES |
| formato cardNumber | A1: formato cardNumber corretto [property validCardNumber] A2: formato cardNumber non corretto |
| valore expiration | B1: valore expiration corretto [property validExpiration] B2: valore expiration non corretto [if validCardNumber] |
| formato cvv | C1: formato cvv corretto [property validCVV] C2: formato cvv non corretto [if validCardNumber & validExpiration] |
| range price | D1: in range D2: non in range [if validCardNumber & validExpiration & validCVV] |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|--------|
| A1-B1-C1-D1 | OK |

| | |
|-------------|------------------------------|
| A2-B1-C1-D1 | Error: return false |
| A1-B2-C1-D1 | Error: return false |
| A1-B1-C2-D1 | Error: return false |
| A1-B1-C1-D2 | Error: "Invalid price range" |

StorageManagement

AddressDAO

doSave

| | |
|----------------------------|--|
| METHOD NAME | doSave |
| PARAMETERS | address: Address |
| ENVIRONMENT OBJECTS | database |
| CATEGORIES | CHOICES |
| Valore address | VA1: valore NON null VA2: valore null |

| | |
|---------------------|---------------------------------|
| COMBINATIONS | ORACLE |
| VA1 | OK |
| VA2 | Error:"Address cannot be null". |

findByKey

| | |
|----------------------------|--|
| METHOD NAME | findByKey |
| PARAMETERS | addressId: Integer |
| ENVIRONMENT OBJECTS | database |
| CATEGORIES | CHOICES |
| Valore addressId | VA1: valore >0[property validId] VA2: valore <=0 |
| Addrss in db | VD1: AdressId associato ad un adress in db[if validId] VD2: AdressId non associato ad un adress in db |

| COMBINATIONS | ORACLE |
|--------------|---|
| VA1-VD1 | OK |
| VA2-VD1 | Error:"Id cannot be negative". |
| VA1-VD2 | Error:"No Address found with the given id". |

findByCustomerId

| METHOD NAME | findByCustomerId |
|---------------------|--|
| PARAMETERS | customerId: Integer |
| ENVIRONMENT OBJECTS | database |
| CATEGORIES | CHOICES |
| Valore customerId | C1: valore >0[property validId] C2: valore <=0 |
| Customer in db | CD1: CustomerId associato ad un customer in db[if validId] CD2: CustomerId non associato ad un customer in db |

| COMBINATIONS | ORACLE |
|--------------|---|
| C1-CD1 | OK |
| C2-CD1 | Error:"CustomerId cannot be negative". |
| C1-CD2 | Corretto: viene restituita una lista vuota. |

ProductDAO

doSave

| METHOD NAME | doSave |
|---------------------|--|
| PARAMETERS | product: Product |
| ENVIRONMENT OBJECTS | database |
| CATEGORIES | CHOICES |
| Valore product | P1: valore non null P2: valore null |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|---------------------------------|
| P1 | OK |
| P2 | Error: "Product cannot be null" |

doUpdate

| | |
|---------------------|--|
| METHOD NAME | doUpdate |
| PARAMETERS | product: Product |
| ENVIRONMENT OBJECTS | database |
| CATEGORIES | CHOICES |
| Valore product | P1: valore non null P2: valore null |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|---------------------------------|
| P1 | OK |
| P2 | Error: "Product cannot be null" |

findByKey

| | |
|---------------------|---|
| METHOD NAME | findByKey |
| PARAMETERS | productId: Integer |
| ENVIRONMENT OBJECTS | database |
| CATEGORIES | CHOICES |
| Valore productId | P1: valore > 0 [property validId] P2: valore <=0 |
| Product in db | PD1: productId associato a un product nel db [if validId] PD2: productId non associato a un product nel db |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|--|
| P1-PD1 | OK: viene restituito il prodotto associato a productId |
| P2-PD1 | Error: "Id cannot be negative" |
| P1-PD2 | Error: "No Product found with the given id." |

findByProdCode

| | |
|----------------------------|---|
| METHOD NAME | findByProdCode |
| PARAMETERS | prodCode: Integer |
| ENVIRONMENT OBJECTS | database |
| CATEGORIES | CHOICES |
| Valore prodCode | P1: valore > 0 [property validCode] P2: valore <=0 |
| Product in db | PD1: prodCode associato a un product nel db [if validId] PD2: prodCode non associato a un product nel db |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|---|
| P1-PD1 | OK: viene restituito il prodotto associato a prodCode |
| P2-PD1 | Error: "prodCode cannot be negative" |
| P1-PD2 | Error: "No Product found with the given prodCode." |

findByCategory

| | |
|----------------------------|---|
| METHOD NAME | findByCategory |
| PARAMETERS | category: Product.Category |
| ENVIRONMENT OBJECTS | database |
| CATEGORIES | CHOICES |
| Valore category | C1: valore non null [property catNotNull] C2: valore null |
| Corrispondenza Categorie | CC1: zero corrispondenze [if catNotNull] CC2: una o più corrispondenze [if catNotNull] |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|--|
| C1-CC1 | OK: viene restituita una lista vuota |
| C2-CC1 | Error: "Category cannot be null" |
| C1-CC2 | OK: viene restituita una lista contenente uno o più prodotti |

doRetrieveAll

| | |
|----------------------------|---|
| METHOD NAME | doRetrieveAll |
| PARAMETERS | |
| ENVIRONMENT OBJECTS | database |
| CATEGORIES | CHOICES |
| Presenza Prodotti | PP1: zero corrispondenze PP2: una o più corrispondenze |

TEST CASES

| | |
|---------------------|---|
| COMBINATIONS | ORACLE |
| PP1 | OK: viene restituita una lista vuota |
| PP2 | OK: viene restituita una lista con uno o più prodotti |

UserDAO

doSave

| | |
|----------------------------|--|
| METHOD NAME | doSave |
| PARAMETERS | user: User |
| ENVIRONMENT OBJECTS | database |
| CATEGORIES | CHOICES |
| Valore user | U1: valore non null U2: valore null |

TEST CASES

| | |
|---------------------|------------------------------|
| COMBINATIONS | ORACLE |
| U1 | OK |
| U2 | Error: "User cannot be null" |

getUserByUsernameAndPassword

| | |
|----------------------------|------------------------------------|
| METHOD NAME | getUserByUsernameAndPassword |
| PARAMETERS | username: String, password: String |
| ENVIRONMENT OBJECTS | database |
| CATEGORIES | CHOICES |

| | |
|--|---|
| Valore username | U1: valore non null [property usernameNotNull] U2: valore null |
| Valore password | P1: valore non null [property passwordNotNull] P2: valore null [if usernameNotNull] |
| Corrispondenza username & password in db | CD1: corrispondenza nel db [if usernameNotNull & passwordNotNull] CD2: nessuna corrispondenza nel db |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|---|
| U1-P1-CD1 | OK |
| U2-P1-CD2 | Error: "Cannot have empty username or password" |
| U1-P2-CD2 | Error: "Cannot have empty username or password" |
| U1-P1-CD2 | Error: "No User found with the given credentials" |

getUserRoles

| | |
|---------------------|--|
| METHOD NAME | getUserRoles |
| PARAMETERS | userId: Integer |
| ENVIRONMENT OBJECTS | database |
| CATEGORIES | CHOICES |
| Valore userId | U1: valore > 0 [property validUserId] U2: valore <= 0 |
| User in db | UD1: user corrispondente in db [if validUserId][property corrisp] UD2: user non corrispondente in db |
| Ruoli associati | NR1: un solo ruolo associato [if validUserId & corrisp] NR2: più ruoli associati [if validUserId & corrisp] |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|---|
| U1-UD1-NR1 | OK: viene restituita una lista con un solo elemento |
| U1-UD1-NR2 | Ok: viene restituita una lista con più di un elemento |
| U2-UD2 | Error: "Id cannot be negative" |
| U1-UD2 | Error: "No User found with the given id" |

findByKey

| | |
|----------------------------|---|
| METHOD NAME | findByKey |
| PARAMETERS | userId: Integer |
| ENVIRONMENT OBJECTS | database |
| CATEGORIES | CHOICES |
| Valore userId | U1: valore > 0 [property validId] U2: valore <=0 |
| User in db | UD1: userId associato a un User nel db [if validId] UD2: userId non associato a un User nel db |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|---|
| U1-UD1 | OK: viene restituito lo User associato a userId |
| U2-UD2 | Error: "Id cannot be negative" |
| U1-UD2 | Error: "No User found with the given id" |

findByUsername

| | |
|----------------------------|---|
| METHOD NAME | findByUsername |
| PARAMETERS | username: String |
| ENVIRONMENT OBJECTS | database |
| CATEGORIES | CHOICES |
| Valore username | U1: username != null [property validUsername] U2: username = null |
| User in db | UD1: username associato a un User nel db [if validUsername] UD2: username non associato a un User nel db |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|---|
| U1-UD1 | OK: viene restituito lo User associato a username |
| U2-UD2 | Error: "Username cannot be null" |
| U1-UD2 | Error: "No User found with the given username" |

CustomerDAO

doSave

| | |
|----------------------------|--|
| METHOD NAME | doSave |
| PARAMETERS | customer: Customer |
| ENVIRONMENT OBJECTS | database |
| CATEGORIES | CHOICES |
| Valore customer | C1: valore non null C2: valore null |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|----------------------------------|
| C1 | OK |
| C2 | Error: "Customer cannot be null" |

findByKey

| | |
|----------------------------|--|
| METHOD NAME | findByKey |
| PARAMETERS | customerId: integer |
| ENVIRONMENT OBJECTS | database |
| CATEGORIES | CHOICES |
| Valore customerId | C1: valore>0 C2: valore<=0 |
| Customer iin db | CD1:CustomerId associato a un Customer nel db CD2:CustomerId non associato a un Customer nel db |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|---|
| C1-CD1 | OK |
| C2-CD1 | Error: "Customer ID cannot be negative" |
| C1-CD2 | Error:"No Customer found with the given id" |

findAddressByCustomerId

| | |
|--------------------|-------------------------|
| METHOD NAME | findAddressByCustomerId |
|--------------------|-------------------------|

| | |
|----------------------------|--|
| PARAMETERS | customerId: integer |
| ENVIRONMENT OBJECTS | database |
| CATEGORIES | CHOICES |
| Valore customerId | C1: valore>0 C2: valore<=0 |
| Customer in db | CD1:CustomerId associato a un Address nel db CD2:CustomerId non associato a un Address nel db |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|---|
| C1-CD1 | OK |
| C2-CD1 | Error: “Customer ID cannot be negative” |
| C1-CD2 | Corretto: viene restituita una lista vuota. |

CustomerOrderDAO

doSave

| | |
|----------------------------|--|
| METHOD NAME | doSave |
| PARAMETERS | order: CustomerOrder |
| ENVIRONMENT OBJECTS | database |
| CATEGORIES | CHOICES |
| Valore order | O1: valore non null O2: valore null |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|---------------------------------------|
| O1 | OK |
| O2 | Error: “CustomerOrder cannot be null” |

updateStatus

| | |
|----------------------------|---|
| METHOD NAME | updateStatus |
| PARAMETERS | orderId: Integer, status CustomerOrder.Status |
| ENVIRONMENT OBJECTS | database |

| CATEGORIES | CHOICES |
|---------------------|---|
| Valore orderId | O1: valore > 0 [property validId] O2: valore <=0 |
| Valore status | S1: valore non null [property validStatus] S2: valore null [if validId] |
| CustomerOrder in db | PD1: orderId associato a un ordine nel db [if validId] PD2: orderId non associato a un ordine nel db |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|---|
| O1-S1-PD1 | OK |
| O2-S1-PD2 | Error: "Id cannot be negative" |
| O1-S2-PD1 | Error: "Status cannot be null" |
| O1-S1-PD2 | Error: "No CustomerOrder found with the given id" |

findByKey

| METHOD NAME | findByKey |
|---------------------|---|
| PARAMETERS | id: Integer |
| ENVIRONMENT OBJECTS | database |
| CATEGORIES | CHOICES |
| Valore id | I1: valore > 0 [property validId] I2: valore <=0 |
| Product in db | PD1: id associato a un CustomerOrder nel db [if validId] PD2: id non associato a un CustomerOrder nel db |

TEST CASES

| COMBINATIONS | ORACLE |
|--------------|---|
| I1-PD1 | OK: viene restituito il prodotto con id specificato |
| I2-PD2 | Error: "Id cannot be negative" |
| I1-PD2 | Error: "No CustomerOrder found with the given id" |

doRetrieveAll

| | |
|-------------|---------------|
| METHOD NAME | doRetrieveAll |
| PARAMETERS | |

| | |
|----------------------------|---|
| ENVIRONMENT OBJECTS | database |
| CATEGORIES | CHOICES |
| Presenza CustomerOrder | PC1: zero corrispondenze PC2: una o più corrispondenze |

TEST CASES

| COMBINATIONS | ORACLE |
|---------------------|---|
| PC1 | OK: viene restituita una lista vuota |
| PC2 | OK: viene restituita una lista con uno o più prodotti |

Test schedule

Abbiamo deciso di dare priorità al test di unità sul livello di business(dato che la maggior parte della logica di business risiede all'interno di tale livello). Abbiamo quindi effettuato il test di unità prima sulle entity e successivamente sulle classi DAO, utilizzando i mock per rappresentare eventuali dipendenze.

Siamo poi passati al test di integrazione, dove abbiamo utilizzato gli stessi casi di test del test di unità, eliminando i mock e utilizzando un approccio bottom up.

Infine abbiamo eseguito il functional testing.

Gli esiti dei test ed eventuali correzioni apportate al sistema in seguito a delle failure, sono state riportate all'interno del documento TER (Test Execution Report).