



iSnapGaming

TP

Introduction	3
Relationship to other documents	3
System Overview	3
Features to be tested/not to be tested	3
Pass/Fail criteria	4
Approach	4
Tools used for testing	4
Test Cases	4
Functionality Testing	4
Autenticazione	4
Registrazione	5
Aggiunta Prodotto al carrello	6
Modifica quantità prodotto nel carrello	7
Effettuazione acquisto	7
Cambio ruolo	8
Aggiunta Prodotto	8
Unit Testing	10
User	10
Address	11
OrderManager	11
ProductManager	14
Product	16
Cart	17
CustomerOrder	19
OrderCreation	20
PaymentAuthorizationServiceAdapter	20
AddressDAO	21
ProductDAO	23
UserDAO	25
CustomerDAO	28
CustomerOrderDAO	29
Test schedule	31

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

Inoltre, 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

Di seguito non sono presenti tutte le funzionalità che devono essere testate, infatti, per le più banali non è stato necessario realizzare molteplici casi d'uso siccome sono delle semplici interazioni con il sistema. Esempi di queste funzionalità sono la modifica dello stato di un ordine e la rimozione di un prodotto.

Autenticazione

PARAMETERS	username, password
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]
--------------------------------	---

TEST CASES

COMBINATIONS	ORACLE
UP1-PP1	Utente autenticato e reindirizzato alla corrispondente homepage.
UP2-PP1	Visualizzazione errore "Wrong username or password"
UP1-PP2	Visualizzazione errore "Wrong username or password"

Registrazione

PARAMETERS	firstName, lastName, dateOfBirth, email, password, conferma password, street, city, postalCode
OBJECTS OF THE ENVIRONMENT	Database
CATEGORIES	CHOICES
Formato firstName (sole lettere)	F1: firstName rispetta il formato del nome [property firstNameVal] F2: firstName non rispetta il formato del nome
Formato lastName (sole lettere)	L1: lastName rispetta il formato del cognome [property lastNameVal] L2: lastName non rispetta il formato del cognome [if firstNameVal]
Valore dateOfBirth (non può essere nel futuro)	D1: dateOfBirth corretta [property dateOfBirthVal] D2: dateOfBirth non corretta [if firstNameVal & lastNameVal]
Formato email per username	E1: email corretta [property emailVal] E2: email non corretta [if firstNameVal & lastNameVal & dateOfBirthVal]
Lunghezza password (>= 8)	LP1: lunghezza password valida [property lenPwVal] LP2: lunghezza password non valida [if firstNameVal & lastNameVal & dateOfBirthVal & emailVal]
Formato password (maiuscole, minuscole, cifre)	P1: password rispetta il formato [property validPwFormat] P2: password non rispetta il formato [if firstNameVal & lastNameVal & dateOfBirthVal & emailVal & lenPwVal]
Valore confirmPassword	CP1: Conferma password corrisponde a password [property pwMatch] CP2: Conferma password non corrisponde a password [if firstNameVal & lastNameVal & dateOfBirthVal & emailVal & lenPsVal & validPwFormat]
Formato City (solo caratteri)	C1: city rispetta il formato city [property cityVal] C2: city non rispetta il formato city [if firstNameVal & lastNameVal & dateOfBirthVal & emailVal & lenPsVal & validPwFormat & pwMatch]

Formato postalCode (len = 5, solo cifre)	PC1: postalCode rispetta il formato postalCode [property postalCodeVal] PC2: postalCode non rispetta il formato postalCode [if firstNameVal & lastNameVal & dateOfBirthVal & emailVal & lenPsVal & validPwFormat & pwMatch]
Username non presente nel database	U1: Username non presente nel database U2: Username presente nel database [if firstNameVal & lastNameVal & dateOfBirthVal & emailVal & lenPsVal & validPwFormat & pwMatch & postalCodeVal]

TEST CASES

COMBINATIONS	ORACLE
F1-L1-D1-E1-LP1-P1-CP1-C1-PC1-U1	Utente registrato correttamente.
F2-L1-D1-E1-LP1-P1-CP1-C1-PC1-U1	Visualizzazione errore "First name must contain only letters"
F1-L2-D1-E1-LP1-P1-CP1-C1-PC1-U1	Visualizzazione errore "Last name must contain only letters"
F1-L1-D2-E1-LP1-P1-CP1-C1-PC1-U1	Visualizzazione errore: "Wrong date format"
F1-L1-D1-E2-LP1-P1-CP1-C1-PC1-U1	Visualizzazione errore "Email entered not in correct format"
F1-L1-D1-E1-LP2-P1-CP1-C1-PC1-U1	Visualizzazione errore "Password must contain at least one number, one uppercase and lowercase letter, and at least 8 or more characters"
F1-L1-D1-E1-LP1-P2-CP1-C1-PC1-U1	Visualizzazione errore "Password must contain at least one number, one uppercase and lowercase letter, and at least 8 or more characters"
F1-L1-D1-E1-LP1-P1-CP2-C1-PC1-U1	Visualizzazione errore "Passwords do not match"
F1-L1-D1-E1-LP1-P1-CP1-C2-PC1-U1	Visualizzazione errore "The name of the city must be composed of only letters"
F1-L1-D1-E1-LP1-P1-CP1-C1-PC2-U1	Visualizzazione errore "The postal code must have only 5 numbers"
F1-L1-D1-E1-LP1-P1-CP1-C1-PC1-U2	Visualizzazione errore "That user already exists"

Aggiunta Prodotto al carrello

PARAMETERS	prodotto, quantity
OBJECTS OF THE ENVIRONMENT	
CATEGORIES	CHOICES
Valore prodotto	P1:Prodotto disponibile [property ProdVal] P2:Prodotto non disponibile

Valore quantità	Q1: quantità selezionata disponibile [property quantityVal] Q2: quantità selezionata non disponibile [if ProdVal]
-----------------	--

TEST CASES

COMBINATIONS	ORACLE
P1-Q1	prodotto aggiunto correttamente al carrello
P1-Q2	Visualizzazione errore “The selected quantity is not available”
P2-Q1	Viene visualizzato il messaggio: “Product temporarily unavailable”

Modifica quantità prodotto nel carrello

PARAMETERS	prodotto, quantity
OBJECTS OF THE ENVIRONMENT	
CATEGORIES	CHOICES
Valore prodotto	P1: Prodotto nel carrello [property prodInCart]
Valore quantità	Q1: nuova quantità disponibile [if prodInCart] Q2: nuova quantità non disponibile [if prodInCart]

TEST CASES

COMBINATIONS	ORACLE
P1-Q1	Quantità del prodotto modificata correttamente nel carrello
P1-Q2	Visualizzazione errore “The selected quantity is not available”

Effettuazione acquisto

PARAMETERS	indirizzo, numero carta, scadenza, CVV
OBJECTS OF THE ENVIRONMENT	Database
CATEGORIES	CHOICES
Valore indirizzo	I1: Indirizzo != null [property addressNotNull] I2: Indirizzo = null
Formato numero carta (NNNN-NNNN-NNNN-NNNN, dove N è una cifra compresa tra 0 e 9)	NC1: numero carta rispetta il formato [property validCard] NC2: numero carta non rispetta il formato [if addressNotNull]
Valore scadenza (non può essere nel passato)	S1: scadenza corretta [property validExpiration] S2: scadenza non corretta [if addressNotNull & validCard]

Formato CVV (deve avere 3 cifre)	C1: CVV rispetta il formato [property validCVV] C2: CVV non rispetta il formato [if addressNotNull & validCard & validExpiration]
Pagamento ha avuto successo	P1: Il pagamento ha avuto successo [if addressNotNull & validCard & validExpiration & validCVV] P2: Il pagamento non è andato a buon fine

TEST CASES

COMBINATIONS	ORACLE
I1-NC1-S1-C1-P1	Il pagamento è andato a buon fine
I1-NC1-S1-C1-P2	Il pagamento non è andato a buon fine
I2-NC1-S1-C1-P2	Visualizzazione errore "Please, insert a valid address"
I1-NC2-S1-C1-P2	Visualizzazione errore "The card format must contain only numbers and in NNNN-NNNN-NNNN-NNNN format"
I1-NC1-S2-C1-P2	Visualizzazione errore "You cannot use an expired card"
I1-NC1-S1-C2-P2	Visualizzazione errore "The security number must contain only numbers and in NNN format"

Cambio ruolo

PARAMETERS	ruolo
OBJECTS OF THE ENVIRONMENT	
CATEGORIES	CHOICES
ruolo	R1: ruolo != null R2: ruolo = null

TEST CASES

COMBINATIONS	ORACLE
R1	OK
R2	Visualizzazione errore "Please select a role"

Aggiunta Prodotto

PARAMETERS	prodCode, name, softwareHouse, platform, price, quantity, category, pegi, releaseYear, imagePath
------------	--

OBJECTS OF THE ENVIRONMENT	
CATEGORIES	CHOICES
Valore prodCode (>0)	PC1: prodCode != null [property prodCodeVal] PC2: prodCode = null
Valore nome	N1: nome != null [property nameNotNull] N2: nome= null [if prodCodeVal]
Valore softwareHouse	S1: softwareHouse != null [property softHouNotNull] S2: softwareHouse = null [if prodCodeVal & nameNotNull]
Valore platform	P1: platform != null [property platfNotNull] P2: platform = null [if prodCodeVal & nameNotNull & softHouNotNull]
Valore price (>0)	PR1: price valido [property priceVal] PR2: price non valido [if prodCodeVal & nameNotNull & softHouNotNull & platfNotNull]
Valore quantity (>0)	Q1: quantità valida [property quantityVal] Q2: quantità non valida [if prodCodeVal & nameNotNull & softHouNotNull & platfNotNull & priceVal]
Valore category	C1: category != null [property categtnotNull] C2: category = nul [if prodCodeVal & nameNotNull & softHouNotNull & platfNotNull & priceVal & quantityVal]
valore pegi	PG1: pegi != null [property pegiNotNull] PG2: pegi = null [if prodCodeVal & nameNotNull & softHouNotNull & platfNotNull & priceVal & quantityVal & categtnotNull]
valore releaseYear (non deve essere nel futuro)	R1: releaseYear corretta [property relYearVal] R2: releaseYear non corretta [if prodCodeVal & nameNotNull & softHouNotNull & platfNotNull & priceVal & quantityVal & categtnotNull & pegiNotNull]

TEST CASES

COMBINATIONS	ORACLE
PC1-N1-S1-P1-PR1-Q1-C1-PG1-R1	Prodotto aggiunto correttamente all'elenco dei prodotti.
PC2-N1-S1-P1-PR1-Q1-C1-PG1-R1	Visualizzazione errore "Product code cannot be null"
PC1-N2-S1-P1-PR1-Q1-C1-PG1-R1	Visualizzazione errore "Product name cannot be null"
PC1-N1-S2-P1-PR1-Q1-C1-PG1-R1	Visualizzazione errore "Software house cannot be null"
PC1-N1-S1-P2-PR1-Q1-C1-PG1-R1	Visualizzazione errore "Platform cannot be null"
PC1-N1-S1-P1-PR2-Q1-C1-PG1-R1	Visualizzazione errore "Price must be greater than 0"

PC1-N1-S1-P1-PR1-Q2-C1-PG1-R1	Visualizzazione errore "Quantity must be greater than 0"
PC1-N1-S1-P1-PR1-Q1-C2-PG1-R1	Visualizzazione errore "Quantity cannot be null"
PC1-N1-S1-P1-PR1-Q1-C1-PG2-R1	Visualizzazione errore "PEGI cannot be null"
PC1-N1-S1-P1-PR1-Q1-C1-PG1-R2	Visualizzazione errore "Release year cannot be in the future"

Unit Testing

User

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

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

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
-----	---

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"

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"

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"

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"

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"

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

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”

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."

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."

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".

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”.

Product

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

TEST CASES

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".

Cart

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
----------	----

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."

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

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".

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

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"

PaymentAuthorizationServiceAdapter

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"

AddressDAO

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

TEST CASES

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

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

TEST CASES

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

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

TEST CASES

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

ProductDAO

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"

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"

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."

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."

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

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

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"

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"

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"

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"

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

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"

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"

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

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"

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"

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"

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).